

DermaSystem

Gestión Informática en Dermatología

Juan Luis Álvarez Herradón

David Arévalo Ruiz

Jesús García Sojo

Carlos Marhuenda Palmón

Proyecto de la Facultad de Informática

Departamento de Ingeniería del Software e Inteligencia Artificial

Universidad Complutense de Madrid



Trabajo Fin de Grado en Ingeniería Informática e Ingeniería del Software

Curso 2013-2014

Director: Gonzalo Pajares Martinsanz

Colaboradora: María Guijarro Mata-García

Colaboradora externa: Isabel Longo Imedio

Índice

Autorización	1
Resumen	2
Palabras clave	2
Abstract	3
Keywords	3
Introduction	4
1. Introducción	8
1.1. Generalidades	8
1.2. Planteamiento del sistema	9
1.3. Análisis de la situación actual	10
1.4. Objetivos	12
1.5. Organización del trabajo	13
1.6. Contribuciones personales	14
1.6.1. Juan Luis Álvarez Herradón	14
1.6.2. David Arévalo Ruiz	16
1.6.3. Jesús García Sojo	19
1.6.4. Carlos Marhuenda Palmón	21
2. Métodos utilizados en tratamiento de imágenes	23
2.1. Binarización mediante detección de umbral	24
2.1.1. Selección del umbral	24
2.1.2. Método de Otsu	25
2.2. Histograma	26
2.2.1. Media	27
2.2.2. Varianza	27
2.2.3. Asimetría	27
2.2.4. Energía	27
2.2.5. Entropía	28
3. Análisis y diseño	28
3.1. Especificación de requisitos	30
3.1.1. Sistema de marcado	30
3.1.2. Control del sistema de marcado	31
3.1.3. Base de datos	31
3.1.4. Autenticación de usuarios	31

3.1.5. Historial de imágenes	31
3.1.6. Carga y eliminación de imágenes	32
3.1.7. Tratamiento de imágenes	32
3.2. Diagrama de casos de uso	32
3.3. Especificación de casos de uso	33
3.3.1. NewPicture	34
3.3.2. AnalyzePicture	34
3.3.3. ViewTreatedPicture	35
3.3.4. DeletePicture	36
3.3.5. ModifyCommentTreatedPicture	36
3.3.6. ShowHistogram	37
3.4. Diseño	38
3.4.1 Definición de Patrones	39
3.4.2 Plataforma y requerimientos derivados para el desarrollo de la aplicación	44
3.4.3. Base de Datos	45
3.5. Diagrama de Clases	47
3.6. Diagramas de secuencia	49
3.6.1. Diagrama de secuencia subir imagen al historial del paciente	49
3.6.2 Diagrama de secuencia de analizar imagen	51
4. Implementación	53
4.1. Tecnologías utilizadas	53
4.1.1. Java	53
4.1.2. Java Swing	53
4.1.3. Java Advanced Imaging	53
4.1.4. JCalendar	54
4.1.5. JFileChooser	54
4.1.6. Hibernate	54
4.1.7. Assembla	55
4.1.8. IDEs	55
4.1.9. Subclipse	56
4.1.10. Trello	56
4.2. Interfaces	56
4.2.1. PatientPictureHistory	56
4.2.2. PictureProcessingView	57
4.2.3. TreatedPictureView	58
4.2.4. PatientViewInfo	58
5. Conclusiones y futuro	59
5.1. Conclusiones	59
5.2. Futuro de la aplicación	60

Conclusions and future work	61
6. Glosario	63
7. Bibliografía	64
Anexo	65
Manual de usuario	65
A1. Historial de imágenes	65
A1.1. Agregar una imagen al historial de un paciente.	66
A1.2. Borrar una imagen del historial.	67
A1.3. Consultar detalles de imagen.	68
A1.4. Analizar imagen.	68
A1.5. Ver el historial de imágenes tratadas.	69
A1.6. Volver a la información del paciente.	70
A2. Analizar imagen de un paciente	71
A2.1. Recortar una imagen	71
A2.2. Procesar una imagen	74
A2.3. Obtener información de la imagen	76
A2.4. Histograma de la imagen	78
A2.5. Guardar imagen	79
A3. Historial imágenes tratadas	81
A3.1. Volver al historial de imágenes.	82
A3.2. Eliminar imagen tratada.	83
A3.3. Consultar detalles de imagen tratada.	83

Autorización

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Firmado:

Juan Luís Álvarez Herradón

David Arévalo Ruiz

Jesús García Sojo

Carlos Marhuenda Palmón

Resumen

DermaSystem es un proyecto desarrollado como Trabajo de Fin de Grado que ha servido para que sus autores demuestren conocimientos adquiridos en la Ingeniería Informática, llevándolos a la práctica, tras sus años de estudio en la Facultad de Informática de la Universidad Complutense de Madrid.

Además les ha dado la oportunidad de trabajar para un cliente real, la Doctora en medicina D^a Isabel Longo Imedio, especialista en Dermatología Médico-Quirúrgica y Venereología en el Hospital Central de la Defensa Gómez Ulla, quien propuso la iniciativa de desarrollar una aplicación de escritorio en la que apoyar sus labores de consulta con los pacientes, no sólo en la administración de los mismos, sino también en los posibles diagnósticos y tratamientos.

Con tal propósito DermaSystem, que constituye la continuación de un trabajo previo desarrollado durante el curso anterior, se diseña con una doble funcionalidad intrínseca: a) desarrollar un módulo de gestión de la Base de Datos con fines de almacenamiento y recuperación de información (datos de pacientes, gráficos de diagnóstico e imágenes); b) desarrollo de un módulo de tratamiento de imágenes dermatológicas, que se almacenan en la Base de Datos tanto las originales como las procesadas para su posterior recuperación durante el proceso de consulta de información de pacientes. El principal objetivo fue llegar a cubrir todas las necesidades y requisitos para dicha aplicación, ampliando los ya existentes, con nuevas funcionalidades como el procesamiento de imágenes.

La ampliación ha consistido exactamente en la mejora del módulo de gestión de la Base de Datos y el desarrollo completo del módulo de tratamiento de imágenes. Todo sin olvidar la gran importancia de hacer de DermaSystem una aplicación escalable, que pueda ser mejorada y ampliada en el futuro, expandiendo su funcionalidad con fin de convertirse en una herramienta imprescindible en los diagnósticos.

Palabras clave

Aplicación informática, sistema de marcado, dermatología, Java, Modelo Vista Controlador, programación orientada por eventos, i18n, Hibernate, tratamiento de imágenes, umbral, histograma, método de Otsu.

Abstract

DermaSystem is a project developed as Final Degree Project that has allowed its authors to demonstrate the knowledge acquired in Computer Science, putting it into practice, after years of study in Facultad de Informática, at Universidad Complutense de Madrid.

In addition, it has brought the opportunity to work for an actual client, the Medicine Doctor Ms. Isabel Longo Imedio, specialist in Medical-Surgical Dermatology and Venereology at Hospital Central de la Defensa Gómez Ulla, in Madrid, who suggested the initiative of developing a desktop application to support her work in patient consultation. Not only to have them managed, but also with possible diagnostics and treatments.

With that purpose DermaSystem, which is the extension of a previous work developed during the last course, is designed with a twofold functionality: a) development of a database management module to store and recover information (patients' data, diagnostic graphics and images); b) developments of a module to process dermatologic images, stored in the database, both the original ones and the processed, so they can be recovered during the consults with patients. The main aim was to cover all needs and requirements for this application, extending the previous ones, with new functionalities like image processing.

The extension has consisted in the database module improvement and the complete development of the new image processing module. All this without forgetting the immense importance of making DermaSystem scalable, getting improved and extended in the future, expanding its functionality in order to become an essential tool in diagnostic medical.

Keywords

Computer-based application, marking system, dermatology, Java, Model View Controller, event-driven programming, i18n, Hibernate, image treating, threshold, histogram, Otsu's method.

Introduction

This project is an extension of the Final Degree Project started by three of our mates in 2012-2013 course (Llorente-Navarro and col., 2013). Called DermaSystem, it was suggested by Medicine Doctor Ms. Isabel Longo Imedio, specialist in Medical-Surgical Dermatology and Venereology at Hospital Central de la Defensa Gómez Ulla, who provided the initiative of developing a desktop application to support her work in patient consultation.

The first version of DermaSystem consisted of Database management with two main roles: administrator and user. Patients' information was arranged in this database, including diagrams where the possible injuries in different parts of the body were marked with proper symbols and descriptions. This system allowed report generations, as well as report recovery for new editions and information updates.

Maintaining the original functionality detailed before, the new version of DermaSystem has evolved towards a new design with new features. In this sense, there are two new improvements. The database management module has been developed with several performances so as to store original images from the patients and also the processed ones, as the information and comments added to each picture. The other new module is able to process the injuries images as a help to diagnostic. The Computer Vision techniques incorporate specific methods from Artificial Intelligence.

As the previous version of DermaSystem, this one is open to new functionalities and improvements related to Database and Image Processing modules in the future.

The design of this application is based on the requirements in the dermatology area and additional considerations of computational sciences developments.

To develop the DermaSystem the following points have been considered:

System approach

The starting point of this project is its real nature, in dermatology, with a specific use in the hospital. It serves as support in daily dermatology practices. The project fulfills with the requirements provided by the specialist as an end-user.

Under these assumptions DermaSystem is designed as a support tool in dermatology, facilitating the daily work. Although DermaSystem has been designed for a specific user, it can be adapted to other medical specialties.

In addition to the above, which is the main objective, the system has been designed considering with a high degree of scalability in order to provide the incorporation of new features and functionalities with minimal effort.

In short, as mentioned before, the project has been focused on the following issues:

- DataBase improvement: addition of new functionalities with reference the previous version.
- Development of a new Image Processing module, with its associated functionalities.
- Integration of both modules, for data and image management.

Current state analysis

To carry out this project, it is vital to know the current status of the area to be covered, management in dermatology by computer-based system, while knowing the dynamic of a consult in dermatology.

As mentioned before, the main goal of the project is to serve as a tool for the specialist, being mandatory to know the actions in the consult. With such purpose, the specialist is the most appropriate to provide this information. Interviews are appropriate tools to extract the underlying knowledge in the area of Knowledge Engineering (Pajares y Santos, 2005).

Considering that the doctor cannot remember each patient's historical record, DermaSystem provides operability in this way, saving time that can be applied to other activities, such as research or diagnosis.

Dermatology is an area where the evolution of injuries is very important, where all details and related data become an important issue.

Objectives

Based on the above, the main goal of DermaSystem is to facilitate the work of the dermatologist related to patient records. Under this assumption the following objective arises:

- Develop a computer-based application to view the logs with maximum ease. The starting point is the previous version of DermaSystem (Llorente-Navarro y col. 2013) with a similar approach and a Database. In the version described in this paper its robustness is to be improved, while new functionalities have been developed to store the information from the images.

In dermatology the evolution of injuries along the time must be studied and analysed. Image processing techniques provide functionalities to facilitate this.

- Develop a new image processing module to extract properties from the injuries. This can be carried by selecting the specific area to be analysed. It is important to guarantee that the area considered matches in all images. It is expected the extension of this module in the future with new functionalities.

In order to add the image processing module, the system must be capable to store the original and processed images, with the corresponding parameters resulting from the diagnosis.

- Modify the Database in the previous version as a result of the addition of the image processing module. This requires the storage of a large number of images.

Because DermaSystem needs to handle protected data form patients, a security system is required as follows:

- User's authentication, with username and password
- Ensure the robustness of the Data-base to preserve the information

Additionally the system must be developed under a friendly interface.

Memory organization

This work is organized in a set of phases, including the architecture and technologies used. Also the objectives are specified to fulfil with the requirements of a real system.

The project has been structured in five phases, where each one contains an essential part concerning the lifetime of a software project:

Requirements specification: to be fulfilled by DermaSystem

Use cases specification: actions during the normal operation in the system

Design: architecture, design patterns, justifying their use

Technologies: use and function in the project

Implementation: design of the software components.

As a previous step, section two includes a brief description of the image processing methods.

The above phases are described in sections three and four, which are the main body of this work. Specifically, section three includes analysis and design of the system, including the requirements, use's cases and design. Section four is devoted to the implementation, including details about the technologies applied. Finally, section five contains the main conclusions and future work.

The user's manual is included as an Annex, containing details about the application and utilities of the different functions available.

1. Introducción

1.1. Generalidades

Este proyecto es la ampliación del Proyecto Fin de Grado comenzado por tres de nuestros compañeros el pasado curso iniciado en el curso 2012-13 (Llorente-Navarro y col., 2013). Tal proyecto, denominado DermaSystem, surgió con la iniciativa de la Doctora en medicina D^a Isabel Longo Imedio, especialista en Dermatología Médico-Quirúrgica y Venereología en el Hospital Central de la Defensa Gómez Ulla, que planteó la necesidad de disponer de una herramienta de gestión y ayuda para el seguimiento y diagnóstico de pacientes en dicha área.

La versión inicial de DermaSystem contenía la gestión de una Base de Datos con dos roles fundamentales, a saber: Administrador y Usuario. En esta Base de Datos se gestionaban los datos de pacientes, incluyendo diagramas ilustrativos donde se marcaban, con la simbología y descripción apropiada, las posibles lesiones de los pacientes en las diferentes partes del cuerpo del paciente. El sistema desarrollado permitía la generación de informes en base a la información almacenada, así como la recuperación de ésta para nuevas ediciones y actualizaciones de la información. De esta forma, cada vez que un paciente acude a consulta, su información tanto de gestión como desde el punto de vista del diagnóstico se actualiza, estando disponible en cualquier momento que se necesite.

Manteniendo la funcionalidad original, descrita previamente, la nueva versión de DermaSystem ha evolucionado hacia un nuevo diseño ofreciendo nuevas prestaciones. En este sentido, son dos las mejoras y ampliaciones realizadas- Por un lado el módulo de gestión de la Base de Datos se ha mejorado considerablemente, incluyendo nuevas prestaciones, particularmente las relativas al almacenamiento de las imágenes originales y procesadas de los pacientes que no poseía la versión inicial. Por otro lado, se ha incorporado un nuevo módulo de Tratamiento de Imágenes, inexistente en la versión inicial, con la finalidad de realizar una serie de procesamientos sobre las imágenes de lesiones de los pacientes como ayuda al diagnóstico. Dichas técnicas, propias de la Visión por Computador, incorporan métodos específicos de Inteligencia Artificial.

Como en el caso de la primera versión de DermaSystem, la que se presenta en este trabajo queda abierta a futuras ampliaciones y mejoras en los dos aspectos mencionados relativos a los módulos Base de Datos y Tratamiento de Imágenes.

Nos encontramos en una época de gran auge tecnológico donde el uso de sistemas informáticos, computadores, tabletas, móviles u otros dispositivos, facilitan grandemente la realización de numerosas tareas. El campo de la medicina no es ajeno a ello. Es conveniente e incluso es posible que en el futuro sea necesario disponer de un sistema que facilite el trabajo de gestión de información de datos de pacientes por

parte de los médicos en general como ayuda al diagnóstico, y más concretamente como en el caso que nos ocupa, de los dermatólogos.

Los médicos en sus consultas y tareas diarias de diagnóstico reciben en consulta numerosos pacientes con patologías diferentes, necesitando disponer de sus datos de manera rápida y sencilla, a fin de poder atenderles de la manera más adecuada y eficaz posible. Cuando los datos están informatizados el personal médico tiene acceso a ellos de forma automática, siendo en este caso necesaria la disponibilidad de herramientas informáticas específicas para cada disciplina, posibilitando así la realización del trabajo de una manera eficaz.

Con vistas a una futura ampliación de DermaSystem con metas más ambiciosas, si cabe, se presenta un diseño modular y flexible de forma que permita dicha ampliación y en su caso mejora, sin descartar su orientación hacia la telemedicina, la cual está en desarrollo y se presenta como una parte muy importante en la medicina del futuro.

Las aplicaciones informáticas requieren de un diseño robusto, a la par que intuitivo y amigable para el usuario, para conseguir que tanto él como el sistema actúen de forma eficiente y segura. En este aspecto, presentamos un diseño que implica el menor esfuerzo posible para llevar a cabo las nuevas tareas que permite realizar el programa.

Por esta serie de razones y motivos, se presenta el diseño de la aplicación informática en dermatología (DermaSystem) cuyo planteamiento y detalles se exponen a lo largo de la presente memoria.

El diseño de esta aplicación está basado en los requisitos del área de la dermatología y bajo las consideraciones de los desarrollos basados en ciencias computacionales.

1.2. Planteamiento del sistema

Uno de los puntos más interesantes del proyecto es su naturaleza real, preparada para un campo específico como es el de la dermatología, con un uso muy concreto dentro del entorno de un hospital.

El planteamiento inicial de la aplicación, materializada en DermaSystem, es el propuesto por la doctora especialista en dermatología, con el fin de poder apoyarse diariamente en la misma para facilitar la realización de su trabajo, concretamente en la tarea de administración de datos, diagnósticos e historiales clínicos de sus pacientes. Este proyecto cumple con los requisitos proporcionados por la doctora, que en esta situación ejerce el rol de cliente y, también, el de usuario final.

Bajo estas premisas y con este propósito se ha desarrollado DermaSystem, cuyo objetivo consiste en ser una herramienta de apoyo para el área de la dermatología. En

concreto lo hará durante las consultas diarias, facilitando las labores rutinarias de manejo de datos que no suponen implicación directa con el paciente o su diagnóstico.

DermaSystem ha sido creado para un cliente concreto, pero se ha querido dotar a la aplicación de cierta escalabilidad. El objetivo es poder ser utilizado por cualquier otro doctor en el campo de la dermatología y servir de inspiración para otras especialidades médicas, dotándole así de un diseño de carácter universal.

Además de lo anterior, que constituye el objetivo principal del trabajo realizado en el proyecto, se plantea un segundo objetivo técnico, no de menor relevancia, planteado en el marco del desarrollo de esta aplicación consistente en facilitar la ampliación y adaptación de las funcionalidades de la misma de acuerdo a las necesidades que vayan surgiendo en el futuro, principalmente desde el punto de vista de los usuarios médicos.

En resumen, según se ha expresado en la sección anterior, el interés del proyecto desarrollado se ha centrado en los siguientes aspectos:

- Mejora del módulo relativo a la Base de Datos existente en la versión original de DermaSystem, con la incorporación de nuevas funcionalidades.
- Desarrollo del nuevo módulo Tratamiento de Imágenes con sus correspondientes funcionalidades.
- Integración de ambos módulos para la gestión de los datos y en particular de las imágenes originales y procesadas.
- Diseño de un sistema robusto con una interfaz amigable para facilitar el uso de la herramienta informática.

1.3. Análisis de la situación actual

Para llevar a cabo este proyecto, resulta de vital importancia conocer el estado actual del área que se quiere cubrir, la gestión en dermatología mediante un sistema informático, a la vez que conocer el funcionamiento de una consulta de esta naturaleza. Como se ha indicado previamente, el objetivo principal de la aplicación es servir como herramienta de apoyo para el médico durante la consulta, por tanto, se han de conocer con el máximo detalle posible todas las rutinas que se pueden llevar a cabo dentro de la propia consulta.

A fin de conocer estos aspectos, no hay mejor forma que consultar a alguien experto en el tema; en nuestro caso contamos con la propia doctora que propuso el proyecto para que nos hiciera saber cuáles son las necesidades que surgen a la hora de revisar los

datos del paciente en consulta, así como los problemas que encuentra al documentar los datos obtenidos durante la mencionada consulta.

Como ya se hiciera en la primera etapa del desarrollo de DermaSystem, las entrevistas constituyen una herramienta de relevancia a la hora de clarificar y explicitar el conocimiento subyacente en el usuario, de forma que desde el punto de vista de la ingeniería informática queden recogidos todos los aspectos relevantes sin omisiones. Este aspecto resulta bien conocido en el ámbito de la ingeniería del software y más específicamente cuando el sistema informático involucra la necesidad de extraer información relevante con implicación de técnicas de conocimiento, tal es el caso de lo que se conoce específicamente como ingeniería del conocimiento (Pajares y Santos, 2005).

A partir de lo anterior, el inconveniente más importante constatado en una consulta es el hecho de que el médico no puede recordar el caso particular de cada paciente, de forma que para llevar a cabo su trabajo, y obtener el mejor diagnóstico posible, necesita conocer el historial de dicho paciente. Leer el historial completo puede resultar una tarea que consume mucho tiempo, alargándose el tiempo de consulta con el consiguiente perjuicio en todos los sentidos, tanto de operatividad como de disminución en la dedicación a otras tareas como puede ser el diagnóstico avanzado o la investigación. Por este motivo, la aplicación desarrollada pretende reducir el tiempo que necesita la realización de ésta y otras tareas.

En un área como el de la dermatología, en la que se debe tener muy en cuenta la evolución de las diferentes lesiones y afecciones del paciente, así como la aparición o desaparición de las mismas. En este sentido, resulta de vital importancia disponer de un sistema que posibilite su consulta y manipulación de forma rápida y eficaz. Más específicamente, y desde el punto de vista del profesional médico se deben conocer con el mayor grado de perfección posible el mayor número de detalles y datos de las lesiones, destacando entre otros el tamaño, el color o la textura, para poder ser tratadas de forma adecuada. Los cambios producidos en el tiempo permiten comprobar la evolución de las mismas respecto a la consulta anterior. Así pues, DermaSystem pretende ser la herramienta ideal para cubrir esta funcionalidad, ofreciendo un sistema de tratamiento y procesamiento de imágenes con el cual, podremos analizar tanto una imagen actual, como la secuencia de imágenes obtenidas en etapas anteriores, cuya finalidad es conocer la evolución de las lesiones a la vez que se caracterizan por sus propiedades.

Teniendo en cuenta lo expuesto anteriormente, a continuación se presentan los objetivos principales que componen el presente proyecto.

1.4. Objetivos

Reiterando lo ya mencionado previamente, la razón por la cual ha surgido la idea de la aplicación DermaSystem no es otra que la de facilitar el trabajo de los médicos, y más concretamente de los dermatólogos, durante las consultas con sus pacientes, resolviendo los problemas que puedan surgir a la hora de obtener el historial del paciente o de documentar el análisis realizado durante la consulta, por lo tanto, se pretende lo siguiente:

- Desarrollar una aplicación informática en dermatología que facilite el tratamiento de la información del paciente, haciendo que el médico pueda revisar el historial médico del paciente durante la consulta, a la vez que añadir las anotaciones convenientes con respecto al tratamiento. Se parte de la versión inicial desarrollada previamente (Llorente-Navarro y col. 2013), que ya contenía un planteamiento y diseño de la Base de Datos. En este caso se trata de mejorar su robustez a la vez que se han diseñado nuevas estructuras para la incorporación de la información procedentes de las imágenes, así como las imágenes mismas.

En un área como la dermatología, donde las lesiones y manchas del paciente aparecen en la piel, por lo que están a simple vista, y que evolucionan a lo largo del tiempo, surge de manera natural la necesidad de analizar y documentar esas lesiones y manchas por medio de imágenes. A su vez, con la integración de los sistemas informáticos en las consultas, resulta natural aprovechar el potencial relativo al procesamiento de imágenes digitales que pueden ofrecer los programas específicos desarrollados a tal efecto, con el fin de simplificar de proporcionar toda la información subyacente posible para facilitar el diagnóstico y su posterior tratamiento. Dicho esto, se plantea:

- Añadir un módulo de tratamiento de imágenes, no desarrollado en la primera versión de DermaSystem, que sirve de apoyo para analizar la evolución de las lesiones y manchas dermatológicas presentes en el paciente. El módulo permitirá seleccionar el área concreta de una imagen a analizar y utilizarla para todas las imágenes de la misma lesión o mancha en el tiempo. Es importante garantizar que el área analizada a lo largo del tiempo sea la misma entre las distintas imágenes que componen la secuencia de una misma zona afectada del paciente. Esto se realiza considerando el tamaño exacto de la lesión o mancha, así como otros datos que permitan conocer el estado de la misma y su evolución a lo largo del tiempo. Se espera la ampliación de este módulo en el futuro con la adhesión de nuevos métodos de tratamiento de imágenes, que permitan extraer mayor información subyacente en las imágenes.

Para añadir el citado módulo de tratamiento de imágenes, surge la necesidad de disponer de un sistema de almacenamiento para las imágenes, tanto las originales como las que hayan sido tratadas y analizadas, así como para todos los parámetros y

resultados diagnósticos obtenidos mediante dicho análisis. Todo esto desemboca en el siguiente objetivo:

- Modificar la Base de Datos de la versión original de DermaSystem para adaptarla a las nuevas necesidades surgidas como consecuencia de añadir el módulo de procesamiento de imágenes. Estas necesidades consisten en almacenar en la Base de Datos un número de imágenes relativamente grande, así como la correspondiente información derivada del análisis de cada una de ellas.

Al ser DermaSystem una aplicación que maneja datos personales de pacientes, necesita de un sistema de seguridad para que únicamente las personas autorizadas para ello puedan tener acceso a tales datos. Por ello se necesita una base de datos robusta y un sistema de autenticación de usuarios mediante usuario y contraseña. Debido a tal necesidad, se especifican los siguientes objetivos:

- Continuar con el sistema de autenticación de usuarios mediante nombre de usuario y contraseña.
- Asegurar la robustez de la base de datos para que la información privada de los pacientes permanezca segura.

En resumen, estos son los objetivos planteados para el desarrollo de DermaSystem con el fin de convertirse en una herramienta de apoyo a la gestión y ayuda al diagnóstico en el área de la dermatología, y de esa manera, facilitar la labor del dermatólogo durante la consulta.

A lo anterior se añade la necesidad de desarrollar el sistema bajo un entorno amigable para simplificar su uso a la vez que robusto para garantizar su eficiencia.

1.5. Organización del trabajo

En esta memoria se presenta de forma detallada cada una de las etapas de desarrollo por las que ha pasado el proyecto. En ellas se puede encontrar la arquitectura y tecnologías utilizadas. También se encuentran detallados los objetivos perseguidos en el desarrollo de DermaSystem para convertirse en una aplicación útil para el propósito planteado y de utilización en un entorno real, surgido de una necesidad igualmente real.

El proyecto se ha estructurado de acuerdo a cinco etapas bien diferenciadas, cada una de ellas representa una parte esencial de lo que se considera el ciclo de vida de un proyecto de desarrollo de software:

- *Especificación de requisitos*: se detallan los requerimientos indispensables que debía cumplir DermaSystem para ser aceptado por el usuario final.
- *Especificación de casos de uso*: donde se precisan las acciones que se pueden llevar a cabo en la aplicación como consecuencia de los procesos de trabajo habituales. Se determinan tanto el camino a seguir para completar la acción, como los usuarios que podrán realizarlas.
- *Diseño*: se especifica la arquitectura a utilizar en la aplicación. Se nombran los patrones utilizados y se explica cómo y porqué han sido utilizados.
- *Tecnologías utilizadas*: se detallan las tecnologías utilizadas y su función dentro del proyecto.
- *Implementación*: se explica el diseño de los distintos componentes de la aplicación, aportando parte del código desarrollado en forma de imágenes y esquemas.

Como paso previo al desarrollo de las anteriores etapas, se plantea el capítulo dos, relativo a la descripción de los métodos aplicados en el tratamiento de imágenes.

Las etapas anteriores se detallan y concretan a lo largo de las secciones tres y cuatro, que componen una parte importante de la presente memoria. Más específicamente en la sección tres se incluyen los aspectos relativos al análisis y diseño del sistema, donde se contiene el análisis de requisitos, así como los casos de uso y el diseño. La sección cuatro recoge los aspectos relacionados con la implementación. En ellos se incluyen los detalles relativos a las diferentes tecnologías utilizadas. Finalmente la sección cinco contiene las conclusiones más relevantes y el trabajo futuro de cara a las posibles mejoras de la aplicación.

Se incluye como Anexo el manual de usuario de la aplicación conteniendo los detalles de aplicación y utilidad de las diferentes funciones disponibles.

1.6. Contribuciones personales

1.6.1. Juan Luis Álvarez Herradón

La labor principal de mi contribución al proyecto se centró en el análisis de imágenes, tanto en la parte gráfica como en la parte interna de la aplicación. Así fue decidido en una de las primeras reuniones del grupo, a fin de aligerar la carga cognitiva en el proyecto y centrarse exclusivamente en una parte del mismo. Tras especificar los requisitos de esta ampliación, buena parte del tiempo fue dedicada a la investigación, no solo por formación en tratamiento y procesado de imágenes, sino también en el uso

del paquete Java Advanced Imaging (2014) con el que se han llevado a cabo los procesos pesados en el análisis. Este paquete fue recomendado por los directores del proyecto DermaSystem, puesto que ante las diversas opciones en lenguajes de programación para el análisis de imágenes, lo mejor era desarrollarlo en Java, de manera nativa a la aplicación.

El trato con este paquete fue especialmente complicado, dada la escasez en documentación que, además, databa de algunos años atrás, e incluso llevando a enlaces que ya habían desaparecido de Internet. Por tanto, hubo que crear proyectos paralelos que se centraron exclusivamente en conocer las clases de JAI, cómo aprovechar su potencial, depurar sus procedimientos y resolver conflictos y problemas que surgieron a la hora de utilizar un paquete relativamente poco documentado.

En lo que respecta a formación en análisis de imágenes, partiendo desde cero, invertí tiempo en aprender cómo, y especialmente, por qué se utilizaban diversas operaciones en matrices que contienen una imagen, y qué resultados podía obtener que resultasen de interés para el objetivo principal de esta parte del proyecto. En ello, el profesor Gonzalo Pajares fue de gran ayuda gracias a su experiencia en el campo del tratamiento de imágenes y la Visión por Computador. Parte del trabajo realizado consistió en la comprensión del significado de las rutinas de tratamiento de imágenes aplicadas así como la interpretación de sus resultados en dermatología. Gran parte de esta labor se centró en la realización de diversas pruebas de ensayo y error, no sólo desde el punto de vista de eficiencia de los tratamientos realizados sino de las posibilidades de uso en la aplicación a desarrollar. Más específicamente, analicé las rutinas que se especifican a continuación desde el punto de su especificación y validación: binarización y las transformaciones a escala de grises con y sin JAI, observando qué forma podía ser más adecuada para nuestros objetivos. Tras el análisis, parte de la actividad se centró en considerar el desarrollo de rutinas propias o el uso de las JAI, siempre con tendencia al uso de éstas con el fin de que en el futuro se puedan aplicar los métodos del paquete JAI que se consideren necesarios con vistas a las ampliaciones previstas.

Luego de este tiempo de investigación y formación, mi enfoque se trasladó a la implementación en Java de los algoritmos y procesos que se determinaron para la aplicación, considerando cuál sería la metodología a utilizar por el cliente a la hora de llevar a cabo el análisis de las imágenes, es decir, el flujo por el que debería pasar todo proceso de análisis que el usuario desee realizar. Durante la creación de estas herramientas, aprecié detalles puntuales y significativos al hecho de que no sería sencillo decidir qué valor de umbral habría que aplicar a la imagen al comienzo del análisis para capturar correctamente la forma de la mancha o lesión, problema que resolví con la inserción del método de binarización de Otsu (1979), que aprendí y utilicé en la asignatura de Percepción Computacional, de forma tal que el cliente no tuviera que pensar sin conocimiento previo qué valor proporcionar, haciendo el sistema automático por defecto y sólo cambiándose a manual siempre que la imagen tuviera unos patrones concretos o un color muy similar, es decir, como una ayuda externa opcional.

Para garantizar que los resultados obtenidos eran fiables, tras la implementación de las funciones tanto propias como de las JAI, y además se correspondían con la algoritmia real de la teoría utilicé MATLAB (The Matworks, 2014), aprovechando sus funciones predefinidas y validadas, así como su potencia computacional para generar los mismos procesos usados en el proyecto.

Además de esto, tuve que obtener mediante un desarrollo propio la información que queríamos extraer de la imagen, puesto que la utilización del paquete JAI es demasiado potente o costosa para esta operación. Esto consistió en recorrer la matriz de imagen tratada, y con los píxeles de interés obtener de la imagen original la media, varianza, asimetría, energía y entropía. Con toda esta información, implementé la extracción del histograma de la imagen y, por supuesto, la visualización del mismo en su respectiva vista. Con el resto de valores estadísticos, creé otra vista dedicada a su visualización para cada canal de color, en una tabla, de la misma forma que el área de la lesión dermatológica a tratar procedente de la imagen.

Con toda esta implementación terminada, la vista para el análisis de imágenes resultó no haber respetado desde su naturaleza el Modelo Vista Controlador. Por ello, fui el encargado de arreglar este problema y adaptarlo, añadiendo el código y clases necesarios en vista, controlador y modelo para la creación y captura de los eventos nuevos y la nueva forma de mover la información internamente en el programa. No sólo eso, sino que también tuve que adaptar la posible alternancia de idiomas de la aplicación, tanto en inglés como en castellano, ya que el sistema contempla la posibilidad de utilizar estos dos idiomas. Asimismo, integré el nuevo módulo de análisis en el resto de DermaSystem, enlazando las partes ajenas a él, e implementando las llamadas a métodos que se comunican con la base de datos, creados por el resto de miembros del equipo de desarrollo, que se habían dedicado a esta labor. Con ello, terminé resolviendo algunos de los errores de mi parte dedicada del proyecto, surgidos del entramado de la aplicación.

Para concluir mi contribución a este proyecto, colaboré en la obtención de los diagramas presentes en esta memoria, así como en lo relativo a la documentación del proyecto. Posteriormente, añadí la introducción en idioma inglés, los casos de uso, la descripción de las interfaces y parte del anexo dedicada al análisis de imágenes en el manual de usuario.

1.6.2. David Arévalo Ruiz

Para llevar a cabo las ampliaciones planteadas sobre el proyecto decidimos dividir las tareas en dos grupos principales. Por un lado las tareas asociadas a la ampliación de la base de datos, la conexión desde la aplicación a la base de datos y las interfaces para gestionar el nuevo sistema de imágenes. Por otro lado el tratamiento de imágenes. Yo junto con Carlos me dediqué al primer grupo de tareas.

Mis aportaciones al proyecto se podrían resumir en:

- Resolución de problemas iniciales.
- Análisis del proyecto heredado.
- Desarrollo e implementación.
- Integración de los desarrollos generados a partir de ambos grupos de tareas.
- Corrección de fallos y depuración.
- Labores de documentación.

Resolución de problemas iniciales.

Ya que partíamos de una aplicación heredada lo primero que necesitábamos era conseguir que funcionara todo lo desarrollado hasta ese momento. Para ello necesitábamos reproducir la infraestructura que soportaba la aplicación inicial. En un principio busqué un host online gratuito que permitiera accesos remotos para la base de datos, ya que no teníamos acceso a la usada el año anterior. Tras una búsqueda y pruebas en varios servidores encontré db4free.net.

Una vez encontrada la base de datos modifiqué los archivos de configuración asociados a Hibernate para poder acceder a ella sin problemas. Tras esto al intentar iniciar la aplicación me encontré con el problema de no disponer de las claves de acceso para la aplicación, ya que las existentes en la base de datos estaban encriptadas mediante el algoritmo md5. Deshabilitando el algoritmo de encriptado conseguí acceder al modo de administrador y así crear nuevos usuarios, una vez dados de alta volví a habilitar el cifrado.

Entendimiento del proyecto heredado.

Para poder realizar las ampliaciones requeridas con base en el proyecto inicial, primero fue necesario un proceso de comprensión de la estructura del proyecto heredado, así como de los patrones y tecnologías empleadas. Además de repasar el código, necesité refrescar mis conocimientos sobre los conceptos acerca de los patrones Modelo-Vista-Controlador y Singleton, construcción de consultas SQL, manejo de WindowsBuilder.

Del estudio del código del proyecto heredado obtuve el punto de integración para nuestra ampliación. Dentro de la clase PatientView podríamos integrar las diferentes vistas como Internal Frames sin alterar la estructura original del proyecto.

Desarrollo e implementación.

Durante la fase de implementación de las interfaces de usuario y su lógica de negocio asociada, me encargué de mantener el contacto de manera periódica con la

colaboradora del proyecto María Guijarro con un continuo asesoramiento en todo lo concerniente a la programación en Java.

Al comenzar el desarrollo, primero decidimos reformar la base de datos para incluir las nuevas tablas que nos harían falta. Ambos grupos de desarrollo con el asesoramiento de los directores definimos los requisitos a cumplir por la nueva base de datos. Tras lo cual, Carlos y yo, discutimos diferentes esquemas hasta llegar a una solución de consenso. El diseño de la base de datos resultante es la que se muestra en la *figura 8*.

Para la administración de imágenes que deseábamos integrar en el programa, necesitábamos crear dos interfaces de usuarios que gestionaran los diferentes grupos de imágenes, originales y procesadas. Fui el encargado de desarrollar una versión inicial para ellas, que constaba de un panel central para visualizar la imagen seleccionada y pequeños paneles en el lateral derecho para seleccionar otras posibles imágenes. Se trataba de una vista similar a la que se puede ver actualmente en el diseño final en TreatedPictureView o PictureHistoryView, si bien menos depurada y sin las funcionalidades que éstas aportan.

Tras estas actividades, creé el conjunto de clases para soportar la galería de imágenes tratadas, las cuales incluyen: TreatedPicture (pictureProcessing), TreatedPictureView, JDialogPictureDescription TreatedPicture (model), DaoTreated. Además de incluir modificaciones en las clases: PatientView, Controller y Model.

Integración de los desarrollos generados a partir de ambos grupos de tareas.

Una vez finalizados ambos desarrollos procedimos a integrar el módulo de tratamiento de imágenes en el desarrollo principal. En esta tarea colaboré con Juan Luis para llevar a cabo la integración, de manera similar a como se había hecho con las diferentes galerías de imágenes existentes.

Corrección de fallos y depuración.

Al final del desarrollo tuve la tarea de revisar el código y corregir diversos errores que fuimos detectando tras la finalización de la fase de integración. Se trataban de diferentes “bugs” que el programa podría generar en secuencias de ejecución menos habituales.

Labores de documentación.

En cuanto a la elaboración de la memoria, colaboré en la descripción de los casos de uso, en el punto de implementación así como en la obtención de los diferentes diagramas explicativos e imágenes de ejemplo.

Actividades previas

Al comienzo de un proyecto como este, lo más importante es la organización, saber qué tareas hay que realizar y entonces, dividir el trabajo entre los componentes del grupo. En nuestro caso, realizamos diversas reuniones entre los componentes del equipo y los directores del presente trabajo.

En estas reuniones se analizaban los progresos realizados y las futuras acciones a seguir. En las primeras reuniones se establecieron dos campos de trabajo, uno incluía la renovación y adaptación de la Base de Datos a las nuevas necesidades del proyecto y la creación de una interfaz que permitiese el trabajo con imágenes reales de afecciones dermatológicas de los pacientes, así como la gestión de un historial de las mismas. El otro campo de trabajo estaba orientado hacia la investigación en la librería de tratamiento de imágenes JAI, Java Advanced Imaging, la utilización de la misma para aplicar diferentes métodos de tratamiento de imágenes y la implementación de la interfaz que permitiera la realización del análisis de imágenes dentro de la aplicación. Mientras que David y Carlos se hacían cargo del primer campo, Juan Luis y yo comenzamos a trabajar en el segundo.

Investigación

El primer paso consistió en una investigación conjunta sobre la librería JAI, para ello, comenzamos buscando en la página oficial de las JAI (2014), no sin ciertas dificultades, por la ausencia de actualización de la información disponible, ya que la última correspondía al año 2007 y la mala navegación entre los enlaces (páginas no disponibles). Finalmente, encontramos un tutorial sobre la librería que nos sirvió para estudiar los diferentes métodos disponibles y su posible aplicación para la implementación de los métodos de tratamiento de imágenes que DermaSystem iba a proporcionar.

Desarrollo e implementación

Tras la investigación sobre las JAI, librería que no habíamos utilizado anteriormente en el Grado de Ingeniería Informática, realizamos la primera de las reuniones con el director del proyecto, Gonzalo Pajares, experto en el campo del tratamiento de imágenes, para concretar qué métodos de tratamiento de imágenes formarían parte de la aplicación, y de esta manera, diseñar también la interfaz de usuario para el tratamiento de las imágenes. Mientras Juan Luis se encargaba de la implementación de los métodos, yo comencé con la interfaz de usuario. Mantuvimos reuniones semanales, con los directores del proyecto para verificar el progreso y establecer nuevas pautas de actuación de cara a la solución de los problemas surgidos hasta el momento, así como en lo relativo a la continuación del proyecto.

La interfaz, `PictureProcessingView`, consta de un panel en el que se muestra una imagen, y dos grupos de controles, uno para la navegación general dentro de la aplicación, y otro para el propio tratamiento de la imagen. El panel contenedor de la imagen, debía cumplir dos funciones, motivo por el cual se utilizan dos paneles diferentes, un `Selector` y un `DisplayJAI`, que se muestran según la función a cumplir en ese momento. El `Selector` proporciona al usuario la posibilidad de recortar la imagen. Para realizar el recorte, el usuario debe clicar sobre una parte de la imagen y desplazar el ratón sin soltarlo hasta tener seleccionada el área deseada. El `DisplayJAI` muestra tanto la imagen original, recortada o sin recortar, como la imagen obtenida después de realizar el tratamiento correspondiente.

El grupo de controles que dirige las tareas que puede realizar el usuario se compone de lo siguiente, cuyo diseño e implementación corrió a mi cargo:

- Botón de recortar (icono de tijeras), habilita la opción de recortar la imagen.
- Botón de guardar (icono de disquete), guarda tanto el recorte realizado sobre la imagen como el resultado obtenido tras su análisis.
- Botón de restaurar original, permite restaurar la imagen original después de realizar un recorte o tras el análisis.
- Botón de analizar, realiza el tratamiento sobre la imagen.
- Barra y cuadro de texto, permiten cambiar el umbral aplicado en el tratamiento de la imagen.
- Botón de información, muestra los datos obtenidos tras el tratamiento de la imagen.
- Botón de histograma, muestra el histograma de la imagen.

Integración

Al finalizar el trabajo que cada uno de los grupos tenía asignado, se realizó la integración del módulo de tratamiento de imágenes en el bloque principal de `DermaSystem`. Con esto se dio por finalizado el código así como la fase de integración. Tareas en las que participé en las partes mencionadas previamente.

Documentación

Tras la finalización de las partes esenciales de desarrollo e implementación, participé en la documentación de las mismas y recogidas en la presente memoria, en particular en lo que se refiere a la descripción detallada de los métodos que desarrollé.

1.6.4. Carlos Marhuenda Palmón

El proyecto por sus grandes dimensiones fue fraccionado en dos secciones. Una parte relativa al desarrollo de la aplicación en la plataforma java y otra más centrada en el tratamiento de imágenes.

Yo fui asignado a la parte correspondiente a la plataforma java sin perder de vista la del tratamiento de imágenes. Las actividades que realicé durante el desarrollo del proyecto tuvieron una serie de hitos que describo a continuación:

- Entendimiento del proyecto heredado.
- Planteamiento de objetivos reales a afrontar en consenso con los profesores directores.
- Reparto de tareas: diseño e implementación.
- Reuniones periódicas para trasvase de conocimiento y estado de cada una de las partes.
- Memoria del proyecto

Comprensión del proyecto heredado.

El hecho de retomar un proyecto en el que no había colaborado requirió mucha paciencia y esfuerzo por mi parte desde un principio. Uno de mis objetivos iniciales fue entender el proyecto en su globalidad, si bien por la envergadura del mismo tuve que subdividirlo en dos partes.

Base de datos: analizando el modelo relacional de la misma y los atributos declarados pude llegar a ver cómo afrontar las ampliaciones reales que pudiéramos plantear. Además de tener que aprender el uso y nomenclatura de hibernate para poder implantar las mejoras previstas y decididas en las reuniones iniciales del proyecto.

Modelo Vista Controlador: tomando como referencia el proyecto inicial, a pesar de su envergadura, era posible visualizar claramente las clases pertenecientes a cada una de las secciones de este patrón. Razón por la cual, gran parte de la actividad llevada a cabo en el proyecto se centró en el desarrollo de los componentes relativos a este patrón. El objetivo era impedir la violación de alguna de sus normas con el fin de no perder su modularidad ante posibles ampliaciones futuras.

Planteamiento de objetivos reales a afrontar en consenso con los profesores

Tras entender el alcance del proyecto, mi idea y la compartida con los directores del proyecto, era avanzar un paso más tomando como referencia la incorporación de imágenes reales al proyecto y a su tratamiento, aplicando funciones y métodos

específicos sin perder de vista el cliente final de la aplicación a desarrollar y sus necesidades de usuario especialista en dermatología.

Este objetivo lo vimos posible y abordable, ya que varios de los componentes del grupo, entre los que me incluyo, habíamos cursado la asignatura Percepción Computacional, parte de la cual se orienta hacia el tratamiento de imágenes.

Reparto de tareas: diseño e implementación

Las tareas a desarrollar se asignaron entre los componentes del grupo en función del reparto consensuado y realizado entre los miembros del proyecto, sin perder de vista el proyecto en su totalidad y aportando conocimientos o apoyo a los compañeros en cualquier momento que lo requiriesen.

Por mi parte desarrollé concretamente los siguientes componentes partiendo desde la vista “PatientView”. A continuación se muestra un diagrama de clases donde se plasman los componentes que tuve que diseñar y modificar en lo relativo a la implementación para llevar a cabo la ampliación del proyecto tomando como punto de partida la versión existente de DermaSystem.

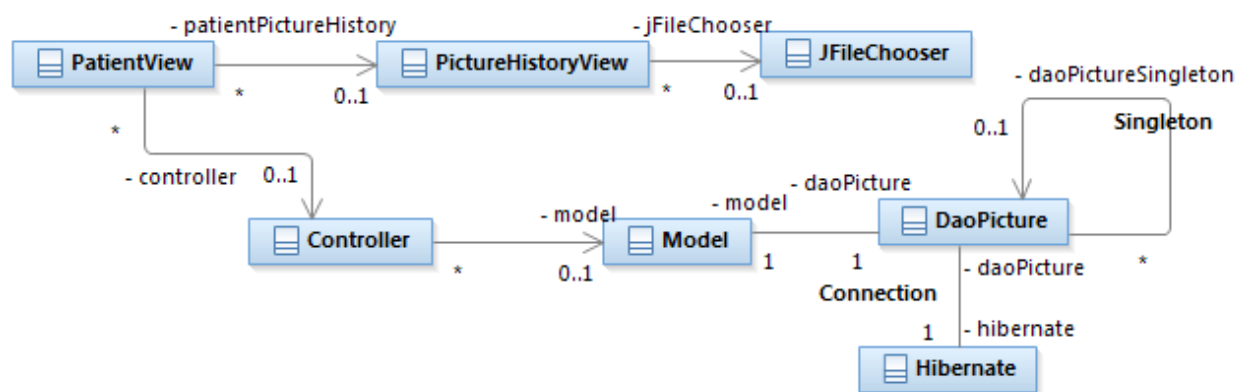


Figura 1: Diagrama de clases

En cuanto a la parte de bases de datos también diseñé junto con David el modelo relacional donde iban a persistir los datos/atributos para la nueva aplicación, quedando finalmente su composición como se muestra en la *figura 8* de esta memoria.

Reuniones periódicas para trasvase de conocimiento y estado de cada una de las partes.

En las reuniones periódicas, habitualmente de carácter semanal entre los miembros del equipo y los directores del proyecto, se analizaba el estado de progreso y los hitos

planteados. Se analizaban los problemas, especificando las soluciones y nuevas tecnologías en su caso.

Memoria del proyecto

Mi participación en este punto ha sido establecer y describir los puntos referentes al diseño e implementación relativos a las tareas asignadas y descritas previamente.

2. Métodos utilizados en tratamiento de imágenes

En un proyecto como el que aquí se presenta, se convierte en una necesidad imperiosa el hecho de sumergirse en campo del tratamiento de imágenes y la visión por computador, relacionadas con las técnicas de inteligencia artificial.

Disponemos de un amplio abanico de posibles tratamientos para aplicar sobre las imágenes, si bien es necesario identificar cuáles son los requeridos para nuestro propósito. En el caso de DermaSystem, el objetivo es obtener los datos más importantes sobre marcas o lesiones que aparecen en las imágenes, para, de esta manera, poder analizar su evolución y como ayuda al diagnóstico médico.

Como resultado de la fase de extracción u obtención del conocimiento llevado a cabo entre los miembros del equipo de desarrollo junto con la doctora en dermatología, se determinan una serie de características y propiedades como las relevantes de cara al tratamiento de las imágenes en relación a los procesos relacionados con las lesiones y manchas dermatológicas. Las características identificadas como esenciales para tal propósito son: tamaño, forma, textura y color. Para obtener los valores relacionados con tales características se determina la necesidad de realizar dos tipos de tratamientos sobre las imágenes, teniendo en cuenta que la unidad básica del tratamiento de las imágenes es el píxel, tales tratamientos se concretan en: locales y globales (Pajares y Cruz, 2007). Los locales se refieren a la consideración de un píxel aislado y a lo sumo teniendo en cuenta sus vecinos más próximos. El color resulta ser una de las características locales más expresivas. Dentro de las operaciones globales se identifican por una parte las relacionadas con la determinación del tamaño de las lesiones y por otra las relativas a la obtención de los descriptores de textura. En el caso global la base del tratamiento lo constituye el histograma de las imágenes, concretamente el histograma relativo a los tres canales, Rojo, Verde y Azul, de que constan las imágenes en el modelo de color RGB (Pajares y Cruz, 2007).

En resumen, los parámetros a analizar se concretan en lo siguiente:

- Color, cuya interpretación resulta inmediata a nivel local.
- Binarización mediante detección de umbral global en la imagen para la determinación del tamaño de las imágenes.

- Histograma, mediante la extracción de propiedades estadísticas globales para el análisis de textura.

A continuación se detallan los aspectos teóricos más relevantes relacionadas con la binarización y el histograma para la determinación de la textura.

2.1. Binarización mediante detección de umbral

El uso de los umbrales en el tratamiento de imágenes constituye una de las principales técnicas en los sistemas de visión industrial para la detección de objetos, especialmente en aplicaciones que requieran procesar una cantidad elevada de datos (Pajares y Cruz, 2007). El objeto de utilización de este método, reside en obtener la región de la imagen definida por la lesión o mancha del paciente, para poder determinar de esta forma el tamaño de la misma.

La manera de poder diferenciar la lesión o mancha de la piel sana, consiste en obtener una imagen binaria, en la que la mancha quedará representada por píxeles negros y la piel sana por píxeles blancos, basándonos en el hecho de que los píxeles de una determinada región presentan una distribución de intensidad similar, por tanto, a partir del histograma de niveles de gris se determina cual es la zona del mismo que corresponde a una determinada región (Pajares y Cruz, 2007). En concreto, se selecciona un nivel que separe los dos tonos de intensidad, conocido como umbral, en inglés *threshold*.

En el caso de DermaSystem, las imágenes de las lesiones y manchas de los pacientes a analizar son en color, concretamente en los tres canales correspondientes al modelo de color RGB. Como paso previo a la binarización de la imagen, se realiza una conversión de la propia imagen a escala de grises mediante la transformación del modelo de color RGB al modelo HSI (matiz, saturación, intensidad). Esta conversión se realiza mediante la transformación apropiada píxel a píxel definida en Pajares y Cruz, (2007).

2.1.1. Selección del umbral

Es muy difícil determinar cuál es el umbral óptimo para poder llevar a cabo una binarización adecuada, por tanto, en DermaSystem se ofrecen dos posibilidades de selección de umbral. La primera consiste en la selección por el método de Otsu, que se describe en el siguiente apartado. La segunda posibilidad consiste en que el usuario especialista en dermatología, de forma manual, puede cambiar el umbral y observar la imagen resultante del cambio. De esta manera podrá elegir el umbral más apropiado para la región seleccionada con el tamaño real de la lesión.

2.1.2. Método de Otsu

El método Otsu (1979) se ha convertido en una técnica ampliamente utilizada, en el mundo del tratamiento de imágenes, para calcular el valor de umbral de forma automática. El desarrollo teórico que sustenta la base del método se describe a continuación.

Los histogramas correspondientes a este tipo de imágenes poseen una característica destacable, es su naturaleza bimodal, formado por dos lóbulos relativamente fácilmente identificables y separables. En un histograma bimodal se supone que el mismo está formado por la suma de dos densidades de probabilidad gaussianas donde cada gaussiana se aproxima a uno de los lóbulos. Este hecho presupone que a medida que las gaussianas se asemejan al histograma real, las desviaciones estándar deben disminuir y como consecuencia de ello se debe elegir aquel umbral que minimice la suma de las varianzas de los dos lóbulos del histograma.

Dada una imagen con L niveles de intensidad y asumiendo que el umbral buscado es T, las probabilidades acumuladas hasta T y desde T hasta L resultan ser,

$$w_1(t) = \sum_{z=1}^T P(z) \quad y \quad w_2(t) = \sum_{z=T+1}^L P(z) \quad (1)$$

A continuación se obtienen las medias y varianzas asociadas:

$$\mu_1(t) = \sum_{z=1}^T zP(z) \quad y \quad \mu_2(t) = \sum_{z=T+1}^L zP(z) \quad (2)$$

$$\sigma_1^2(t) = \sum_{z=1}^T (z - \mu_1(t))^2 \frac{P(z)}{w_1(t)} \quad y \quad \sigma_2^2(t) = \sum_{z=T+1}^L (z - \mu_2(t))^2 \frac{P(z)}{w_2(t)} \quad (3)$$

Finalmente se calcula la varianza ponderada:

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t) \quad (4)$$

Se elige el umbral T correspondiente al nivel de intensidad que proporcione la mínima varianza ponderada definida en la ecuación (4).

En la *figura 2(a)* se muestra una imagen real original con una lesión dermatológica de un paciente rodeada de una zona de piel con otras características. En (b) aparece la imagen binarizada resultante tras la conversión a escala de grises y la aplicación del proceso de umbralización mediante el método de Otsu, donde la lesión aparece en

negro (valor de intensidad 0) y el resto de la piel que la rodea en blanco (nivel de intensidad 255).

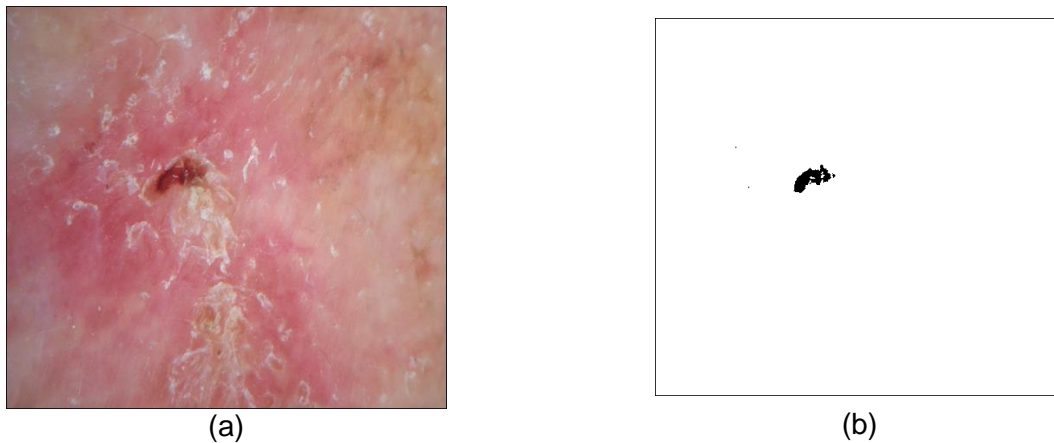


Figura 2: (a) Imagen original; (b) imagen procesada

2.2. Histograma

El histograma de una imagen es una función discreta que representa el número de píxeles en la imagen en función de los niveles de intensidad, g . La probabilidad $P(g)$ de ocurrencia de un determinado nivel g se define como (Pajares y Cruz, 2007),

$$P(g) = \frac{N(g)}{M} \quad (5)$$

donde M es el número de píxeles en la imagen y $N(g)$ es el número de píxeles en el nivel de intensidad g . Como con cualquier distribución de probabilidad todos los valores de $P(g)$ son menores o iguales que 1 y la suma de todos los valores de $P(g)$ es 1.

Además de eso, con el histograma podemos obtener ciertas propiedades estadísticas que resultan de interés para analizar la naturaleza de la textura de la lesión o mancha del paciente y su evolución. Estas propiedades de naturaleza estadística, son las que se describen seguidamente.

2.2.1. Media

Es el valor medio de los niveles de gris y nos informa sobre el brillo general de la imagen, está definida por

$$\bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_i \sum_j \frac{I(i, j)}{M} \quad (6)$$

L es el número total de niveles de gris, así para una imagen con valores de gris entre 0 y 255; L se corresponde con el valor 256. Una imagen brillante tendrá una media alta y viceversa.

2.2.2. Varianza

Mide la dispersión de los valores de intensidad alrededor de la media, está definida por,

$$\sigma^2 = \sum_{g=0}^{L-1} (g - \bar{g})^2 P(g) \quad (7)$$

Una varianza alta corresponde a una imagen con contraste alto y viceversa.

2.2.3. Asimetría

Asimetría sobre la media en la distribución de los niveles de gris,

$$a = \sum_{g=0}^{L-1} (g - \bar{g})^3 P(g) \quad (8)$$

Un valor absoluto alto del parámetro a indica una gran asimetría y al contrario.

2.2.4. Energía

Nos informa sobre la distribución de los niveles de gris:

(9)

$$E = \sum_{g=0}^{L-1} (P(g))^2$$

La energía tiene valor máximo 1 para una imagen con un único nivel de gris y disminuye a medida que aumenta el número de niveles de gris.

2.2.5. Entropía

Nos informa también sobre la distribución de los niveles de gris:

$$e = - \sum_{g=0}^{L-1} P(g) \log_2 [P(g)] \quad (10)$$

Cuanto mayor es el número de niveles de gris en la imagen mayor es la entropía, esta medida tiende a variar inversamente con la energía.

3. Análisis y diseño

Como paso previo a los pasos de análisis y diseño detallados, en la *figura 3* se muestra un esquema general de funcionamiento del sistema, exactamente el mismo que el propuesto en Llorente-Navarro y col. (2013). En dicho diagrama se muestran cuatro bloques básicos que constituyen el conjunto del diseño:

- Interfaz de usuario
- Gestión de la Aplicación
- Base de Datos
- Procesado de imágenes

El usuario accede a la aplicación a través de la interfaz de usuario, que envía información hacia el módulo Gestión de la Aplicación procedente de aquel. Tras la realización de los procesos necesarios el usuario recibe la información pertinente como resultado del proceso realizado.

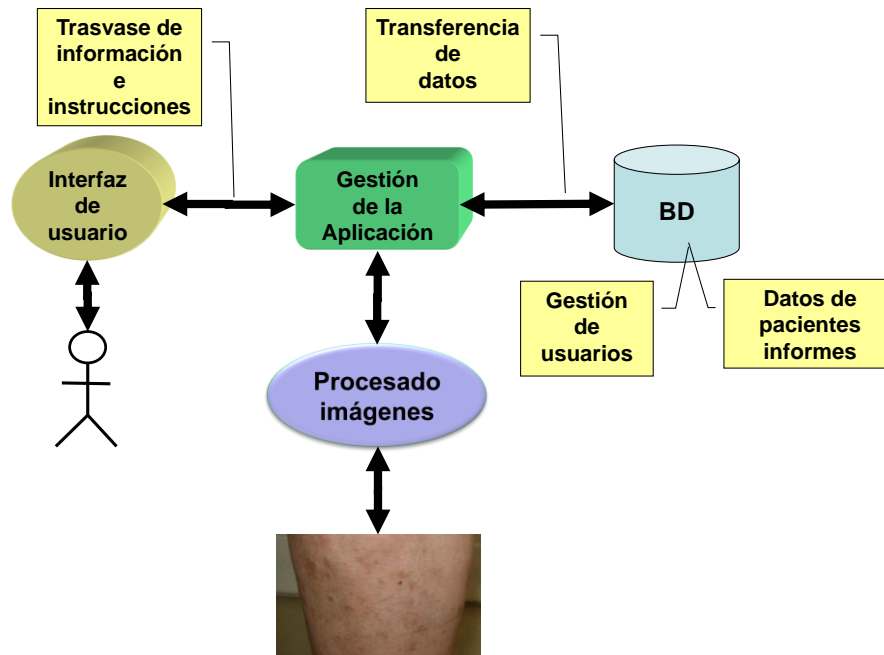


Figura 3: Arquitectura del sistema

La doble flecha determina la existencia de un trasvase de información e instrucciones en el doble sentido especificado.

El módulo Gestión de la Aplicación se encarga de la organización de los procesos, así como de la adquisición de la información procedente bien de la Base de Datos o directamente de las cámaras que capturan las imágenes.

La Base de Datos contiene información relativa tanto a pacientes como a la propia gestión del sistema. En el primer caso se trata de datos e imágenes de pacientes con su correspondiente identificación de informes asociados. En el segundo caso, se hace referencia a la gestión de usuarios por parte del administrador del sistema y el acceso de estos mismos usuarios.

El módulo identificado como procesado de imágenes, que incluye su captura, es el encargado de la obtención de imágenes a través de las cámaras correspondientes, que fotografían las lesiones de los pacientes.

En relación a las fases concretas relativas al análisis y diseño del sistema, éstas se desglosan como sigue:

- *Especificación de requisitos*: que incluye la definición de las funcionalidades del sistema en relación a las necesidades reales del usuario.

- *Especificación de casos de uso*: relativos al establecimiento de la funcionalidad global del sistema.
- *Diseño*: que se concreta en el modelo de patrones utilizados, plataforma de implantación, base de datos.
- *Diagramas de clase y secuencia*: como parte intrínseca del diseño.

3.1. Especificación de requisitos

DermaSystem debe convertirse en la aplicación que, como ya se ha mencionado anteriormente, ayude a los especialistas en dermatología con sus labores diarias en la consulta de cara a la gestión y el diagnóstico de pacientes.

Su base de operación consiste en el manejo apropiado y eficiente de los historiales médicos del paciente, y especialmente con el sistema de marcado y análisis de imágenes, según el esquema mostrado en la *figura 3*.

Por sistema de marcado se entiende la funcionalidad orientada a gestionar la parte del sistema encargada de insertar, editar, modificar y eliminar los símbolos específicamente diseñados con el fin de proporcionar una visualización general de la ubicación de las lesiones y manchas de los pacientes y su distribución en el cuerpo.

La gestión eficiente de los historiales médicos es parte de la base de datos, así como la gestión de usuarios según su rol respecto de la aplicación, esto es administrador del sistema o médico como usuario normal, incluyendo la autenticación de los mismos.

Finalmente, el manejo del historial de las imágenes, la carga y eliminación de las mismas sobre la base de datos, así como la parte relativa al tratamiento de las mismas constituye el tercer aspecto relevante del sistema.

Teniendo en cuenta los aspectos anteriores, la especificación de requisitos del sistema está orientada a recoger los aspectos relevantes relacionados con ellos. A continuación se especifican los requisitos teniendo en cuenta el sistema de marcado y su control, la base de datos en relación al manejo de la información, junto con la autenticación de usuarios. Finalmente se incluyen los aspectos relacionados con las imágenes, su historial, la carga y eliminación, así como su tratamiento.

3.1.1. Sistema de marcado

Este sistema permite a los dermatólogos poder añadir y eliminar marcas, así como crear comentarios específicos para cada una de ellas. Estas marcas se colocarán

sobre imágenes prediseñadas que representen total o parcialmente el cuerpo humano. Además de las marcas, el doctor podrá añadir una descripción detallada propia si lo desea.

3.1.2. Control del sistema de marcado

Las marcas quedarán registradas según la fecha y la hora de su creación. En el caso de la eliminación de las marcas, éstas desaparecerán de la imagen en la que fueron creadas, pero todavía aparecerán como parte del historial del paciente, para poder comprobar la evolución de sus lesiones cuando sea necesario.

3.1.3. Base de datos

Los datos pertenecientes al historial médico del paciente se almacenarán en una base de datos, concretamente se guardarán: los datos personales, todas las marcas y anotaciones realizadas por el doctor, los datos obtenidos y el área concreta de la imagen analizada, así como el diagnóstico realizado asociado a todas o cada una de las marcas identificadas del paciente.

La base de datos necesitará albergar una gran cantidad de datos, así como un elevado número de pacientes, cada uno de ellos a su vez conteniendo un elevado número de imágenes asociadas, es por esto que el diseño de esta base de datos ha sido pensado para ser exportado a un sistema de mayor capacidad y prestaciones, como pueda ser la propia base de datos del Hospital Central de la Defensa Gómez Ulla de Madrid.

3.1.4. Autenticación de usuarios

La aplicación dispondrá de un sistema de verificación de usuarios, el cual permitirá a los propios usuarios acceder adquiriendo diferentes roles en el momento de su registro. El rol de administrador permitirá realizar tareas relacionadas con el mantenimiento de la aplicación gestionando el acceso de los usuarios. El dermatólogo desempeñará el rol de usuario habitual, el cual se limita a utilizar la aplicación para los propósitos para los que ha sido creada.

3.1.5. Historial de imágenes

Resulta interesante disponer de un historial de imágenes para cada paciente, para que el médico pueda observar durante la consulta el estado de las lesiones y marcas en las anteriores visitas del propio paciente. Este historial contendrá cada una de las imágenes almacenadas del paciente, y las relacionadas con cada una de ellas, es decir, las imágenes resultantes obtenidas al analizar tales imágenes. De esta manera podrá realizar comparaciones entre las imágenes actuales, las anteriores y los datos de

las mismas con el fin de determinar la evolución de las lesiones, así como los efectos de los tratamientos seguidos por los pacientes y su consideración.

3.1.6. Carga y eliminación de imágenes

Al introducir en la aplicación tanto el historial como el módulo de tratamiento de imágenes, surge la necesidad de incorporar un sistema para controlar la carga y la eliminación de imágenes en la aplicación. Este sistema permitirá cargar imágenes desde el ordenador en el cual se esté ejecutando la aplicación, y también facilitará la eliminación de las mismas cuando ya no se requiera su conservación. Las imágenes que sean cargadas sobre la aplicación se almacenarán en la base de datos, y se borrarán de la misma al realizar la operación de eliminación.

3.1.7. Tratamiento de imágenes

Las imágenes disponibles de cada paciente, bien por existir en la base de datos o procedentes de una nueva captura podrán ser tratadas con el objetivo de proceder a su análisis y posterior registro de las lesiones o manchas de la manera más detallada posible. De esta forma, se busca facilitar al médico la tarea de comprobar la evolución de la lesión o mancha, permitiendo comparar, mediante valores numéricos proporcionados por las funciones de tratamiento de imágenes, determinados parámetros de interés para el diagnóstico, tales como: tamaño, color, forma o textura.

El médico podrá elegir la zona exacta de la imagen a analizar para obtener unos resultados más precisos resultados del análisis.

3.2. Diagrama de casos de uso

A continuación se muestra el diagrama de *casos de uso* que posteriormente se desglosan en detalle para una mayor comprensión de los mismos. Este diagrama corresponde al subsistema de operaciones o posibles casos que el usuario, dermatólogo en DermaSystem, puede realizar sobre la ficha que contiene todos los datos de un determinado paciente dentro de la aplicación. En la *figura 4* se muestra un diagrama genérico de casos de uso del sistema desarrollado.

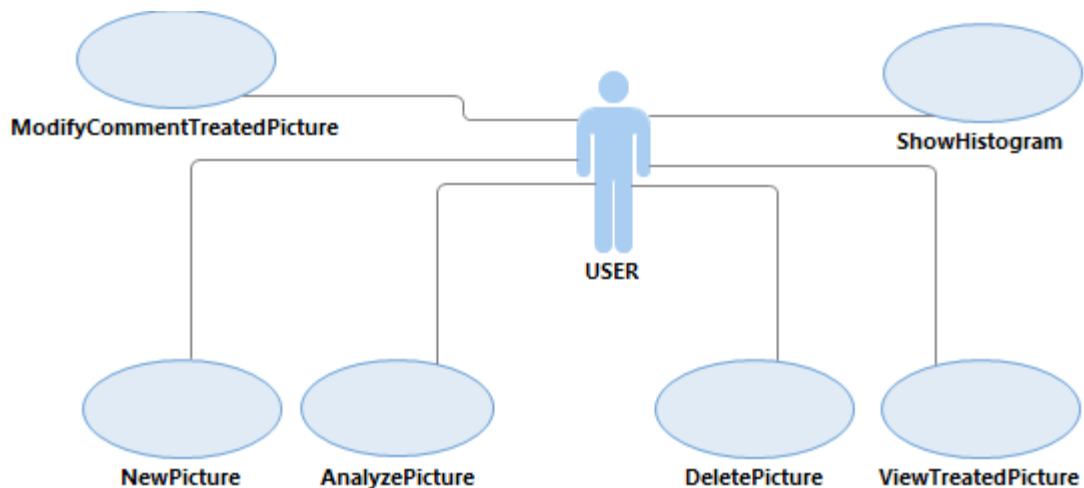


Figura 4: Diagrama genérico de casos de uso

3.3. Especificación de casos de uso

Los casos de uso de los subsistemas de usuarios y pacientes son básicamente los contenidos en la memoria descriptiva de la primera versión de DermaSystem y contenidos en la correspondiente memoria descriptiva (Llorente-Navarro y col., 2013).

En la versión que se describe en la presente memoria sólo existe un carácter de uso y, por consiguiente, un único rol, que es el de usuario estándar o dermatólogo. Si bien en la aplicación completa, además de éste existe, el rol de administrador del sistema, que se desarrolló íntegramente en la primera versión de DermaSystem.

En relación al subsistema de análisis de imágenes, los casos de uso más representativos de dicho subsistema que se han generado son los siguientes:

- **NewPicture:** carga una nueva imagen mediante el selector de archivos y la almacena en la base de datos.
- **AnalyzePicture:** trata una imagen para la posterior extracción de información, y la almacena en la base de datos, relacionada con la imagen original.
- **ViewTreatedPicture:** accede a las imágenes tratadas que están relacionadas con la imagen original.
- **DeletePicture:** elimina una imagen de la base de datos.
- **ModifyCommentTreatedPicture:** modifica los comentarios introducidos al analizar una imagen.

- ShowHistogram: muestra el histograma de la imagen durante el análisis.

A continuación se detallan los casos anteriores, teniendo en cuenta que los relativos al resto del sistema se describen en la mencionada referencia (Llorente-Navarro y col., 2013).

3.3.1. *NewPicture*

Precondiciones:

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio correctamente (*"logueado"*), así como mostrar el paciente a tratar.

Flujo normal:

1. Pulsar el botón "Historial de imágenes"
2. La aplicación muestra la primera imagen del paciente, si la hubiera, así como una galería del resto, todas ellas ordenadas cronológicamente por antigüedad. Pulsando en "Cargar nueva imagen" aparecerá un diálogo de selección de fichero.
3. Seleccionar la imagen y pulsar "Aceptar". Al hacerlo, aparecerá la nueva imagen al final de la galería.

Alternativas:

- Del paso 2 al 3 el doctor puede cancelar el proceso de carga de imágenes.

3.3.2. *AnalyzePicture*

Precondiciones:

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio correctamente (*"logueado"*), debe existir un paciente y una imagen relacionada con el mismo, así como ser seleccionada la imagen a tratar.

Flujo normal:

1. Pulsar el botón "Analizar".

2. La aplicación muestra la imagen en una nueva vista. Pulsar el botón de “Recortar” (tijeras).
3. Seleccionar con el ratón la lesión concreta a analizar, eliminando así la mayor superficie de piel sana posible.
4. Pulsando el botón “Analizar”, la imagen se binarizará apareciendo su nuevo aspecto.
5. Pulsar el botón “Información” para obtener los valores característicos de la imagen.
6. Para guardar el resultado del análisis de la imagen, pulsar “Guardar” (disquete). Aparecerá un diálogo para introducir el nombre. Pulsar “Aceptar”.
7. Aparecerá un diálogo para introducir un comentario. Tras pulsar “Aceptar” se mostrará un diálogo de confirmación.

Alternativas:

- El paso 2 se puede omitir si se considera que la imagen está bien encuadrada con la mancha que se desea tratar.
- Entre los pasos 4 y 5, es posible cambiar el valor automático de umbral. Para ello se procede como sigue:
 - Desactivar el *checkbox* “Umbral automático”.
 - Elegir un valor en el selector deslizante, o introducirlo en el cuadro de texto correspondiente.
- En los pasos 6 y 7 el doctor puede cancelar el proceso de guardado

3.3.3. *ViewTreatedPicture*

Precondiciones:

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio correctamente (“*logueado*”), debe existir un paciente y al menos una imagen asociada al mismo.

Flujo normal:

1. Pulsar el botón “Historial de imágenes”.

2. Seleccionar una imagen de la galería. Tras ello, se mostrará la imagen en el panel central.
3. Pulsar el botón de “Imágenes relacionadas”.
4. Seleccionar la imagen tratada que se desea visualizar.

Alternativas:

- El paso 2 se puede navegar en la galería con los botones “Arriba” y “Abajo”.

3.3.4. DeletePicture

Precondiciones:

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio correctamente (“logueado”), debe haber un paciente y al menos una imagen relacionada al mismo.

Flujo normal:

1. Pulsar el botón “Historial de imágenes”.
2. Seleccionar una imagen de la galería. Tras ello, se mostrará la imagen en el panel central.
3. Pulsar el botón de “Borrar”. Se refrescará la vista con la imagen eliminada.

Alternativas:

- En el paso 3, en caso de error en la eliminación, aparecerá un mensaje de error.

3.3.5. ModifyCommentTreatedPicture

Precondiciones:

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio correctamente (“logueado”), debe haber un paciente, al menos una imagen relacionada con el mismo y una imagen tratada a partir de ésta que se encontrará seleccionada y activa.

Flujo normal:

1. Pulsar el botón “Historial de imágenes”.
2. Seleccionar una imagen de la galería. Tras ello, se mostrará la imagen en el panel central.
3. Pulsar el botón de “Imágenes relacionadas”.
4. Seleccionar la imagen tratada que se desee. Se mostrará esa imagen en el panel central.
5. Pulsar el botón “Detalles”. Aparecerá una ventana emergente con la información obtenida del tratamiento y los comentarios asociados.
6. Pulsar en el cuadro de texto de comentarios y modificar el texto.
7. Pulsar el botón “Guardar”. Desaparecerá la ventana emergente y volverá al estado del punto 4.

Alternativas:

- Durante los pasos 6 y 7 el usuario puede cancelar la acción para no guardar los cambios pulsando en el botón “Cancelar”.
- En el paso 7, en caso de error en la modificación del comentario, aparecerá un mensaje de error.

3.3.6. *ShowHistogram*

Precondiciones:

El usuario que ejecuta la acción debe ser doctor y además estar registrado en el dominio correctamente (“*logueado*”), debe existir un paciente y una imagen relacionada con el mismo, así como ser seleccionada la imagen a tratar.

Flujo normal:

1. Pulsar el botón “Analizar”.
2. La aplicación muestra la imagen en una nueva vista. Pulsar el botón de “Recortar” (tijeras).
3. Seleccionar con el ratón la lesión concreta a analizar, eliminando así la mayor superficie de piel sana posible.

4. Pulsando el botón “Analizar”, la imagen se binarizará y mostrará su nuevo aspecto.
5. Pulsar el botón “Ver histograma”. Aparecerá una ventana emergente con el histograma obtenido a partir del análisis de la imagen.

Alternativas:

- El paso 2 se puede omitir si se considera que la imagen está bien encuadrada con la mancha que se desea tratar.
- Entre los pasos 4 y 5, es posible cambiar el valor automático de umbral. Para ello:
 - Desactivar el *checkbox* “Umbral automático”.
 - Elegir un valor en el selector deslizante, o introducirlo en el cuadro de texto correspondiente.

3.4. Diseño

Aun siendo conscientes de que no es necesario el uso de patrones para desarrollar un software de calidad, lo que si hemos de buscar como desarrolladores de aplicaciones informáticas es la reusabilidad de los módulos del sistema y en su caso del propio sistema en general, cuya finalidad es su utilización en futuros proyectos, ya sea en relación a la extensión del que aquí se presenta o en otros de similar estructura y aplicabilidad. Por ello el diseño contempla el hecho de evitar la reiteración en la búsqueda de respuestas a problemas ya conocidos y solucionados anteriormente, junto con una formalización en forma de un vocabulario común entre diseñadores facilitando la herencia de un proyecto y su comprensión.

Todo lo dicho anteriormente refuerza la necesidad de implantar una estandarización del código, razón por la cual se decide canalizar el diseño a través del uso de patrones, que se traduce en una modularidad de la aplicación de tal forma que nos permita modificar o incluso sustituir cada una de esas partes o módulos con el mínimo esfuerzo y sin dificultad.

3.4.1 Definición de Patrones

Patrón Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC), Reenskaug y Coplien (2009), es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello en el MVC, como su nombre indica, se propone la construcción de tres componentes distintas, a saber: modelo, vista y controlador. Por un lado define componentes para la representación de la información y por otro para la interacción entre el usuario y el sistema. Este patrón de arquitectura de software se basa en las ideas de reutilización de código mediante la separación de conceptos y características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento de la forma más eficiente posible.

Los componentes de MVC, que se muestran en la *figura 5* tomada de Llorente-Navarro y col. (2013), se definen como sigue:

- *Modelo*: es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para ser mostrada. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- *Controlador*: responde a eventos e invoca peticiones al 'modelo' cuando se realiza alguna solicitud sobre la información presentada en la vista. También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo', por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
- *Vista*: presenta el 'modelo', información y lógica de negocio, en un formato adecuado para interactuar (usualmente la interfaz de usuario). Por tanto, requiere de dicho 'modelo' la información que debe representar como salida.

En lo que respecta a la *usabilidad* del patrón en el proyecto, que se describe en la presente memoria, indicar que el desarrollo del mismo ha girado en torno a este patrón permitiendo así la modularidad del mismo y facilitando el reparto de tareas y la asignación de las mismas a cada componente del equipo de desarrollo.

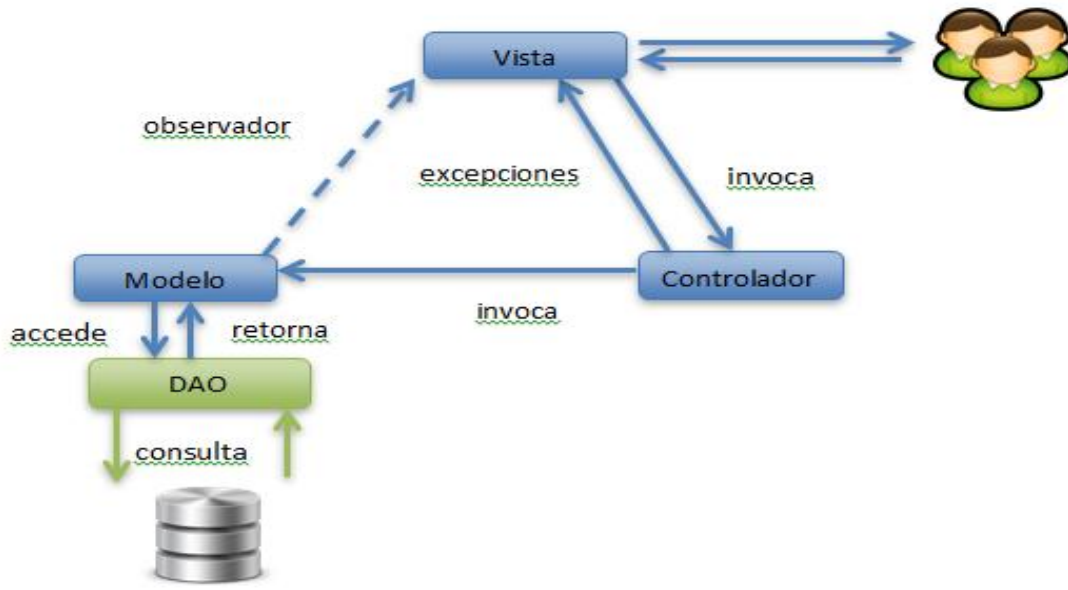


Figura 5: Esquema del patrón MVC

Patrón Observer

Observador (en inglés Observer), Gamma y Col. (2005) es un patrón de diseño que define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. Se trata de un patrón de comportamiento, es decir, está relacionado con algoritmos de funcionamiento y asignación de responsabilidades a clases y objetos.

Los patrones de comportamiento describen no solamente estructuras de relación entre objetos o clases sino también esquemas de comunicación entre ellos.

Ejemplo de clase que implementa Observer:

```
public class PictureHistoryView extends JFrame
implements Observer {

    private static final long serialVersionUID = 1L;

    private Controller controller;
    private Dimension dimension;

    private ArrayList<PictureHistory> relatedPictures;
```

En relación a la *usabilidad* del patrón en el proyecto mencionar que tal y como se puede observar en el esquema de la *figura 5*, a modo de ejemplo, este patrón es

empleado en la parte de la vista para la notificación y actualización de la misma en el caso de que proceda, según los cambios efectuados en la capa de modelo.

Patrón Singleton

El patrón de diseño Singleton (instancia única), Bloch (2008), está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su finalidad consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón Singleton en nuestro proyecto se plasma a través de una clase interfaz denominada DAO que contiene un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor con atributos tales como protegido o privado.

El patrón Singleton provee una única instancia global gracias a que:

- La propia clase es responsable de crear la única instancia.
- Permite el acceso global a dicha instancia mediante un método de clase.
- Declara el constructor de clase como privado para que no se pueda instanciar directamente.

En relación a su usabilidad del patrón en el proyecto, indicar que dicho patrón es empleado en el DAO permitiendo la unicidad del mismo y por tanto el acceso de un único elemento de ese tipo a la base de datos, tal y como se muestra a continuación, en las dos fases siguientes:

- Ejemplo de declaración de la estructura del patrón Singleton en la clase java DaoPicture utilizada en el proyecto.

```
public class DaoPicture implements Dao {  
  
    static private DaoPicture daoSingleton = null;  
  
    public static DaoPicture getDao() {  
        if (daoSingleton == null) {  
            daoSingleton = new DaoPicture();  
        }  
        return daoSingleton;  
    }  
}
```

- Ejemplo de creación del objeto DaoPicture declarado anteriormente como Singleton.

```
public void deletePictureHist(int idPatient, int
idHistoryPicture, String description, Date date) {
    try {
        DaoPicture dao = DaoPicture.getDao();
```

Patrón DAO

Utilizar un Data Access Object (DAO) nos permite abstraer y encapsular todos los accesos a la fuente de datos.

El DAO implementa el mecanismo de acceso requerido para trabajar con la fuente de datos. Esta fuente de datos puede ser un almacenamiento persistente tal como una RDBMS, un servicio externo como un intercambio B2B, un repositorio LDAP o un servicio de negocios al que se accede mediante CORBA Internet Inter-ORB Protocol (IIOP) o bien mediante sockets de bajo nivel.

Los componentes de negocio que tratan con el DAO utilizan un interfaz simple expuesto por el DAO para sus clientes. El DAO oculta completamente los detalles de implementación de la fuente de datos a sus clientes. Como el interface expuesto por el DAO no cambia cuando cambia la implementación de la fuente de datos subyacente, este patrón permite al DAO adaptarse a diferentes esquemas de almacenamiento sin que ello afecte a sus clientes o componentes de negocio. Esencialmente, el DAO actúa como un adaptador entre el componente y la fuente de datos, *figura 6*.

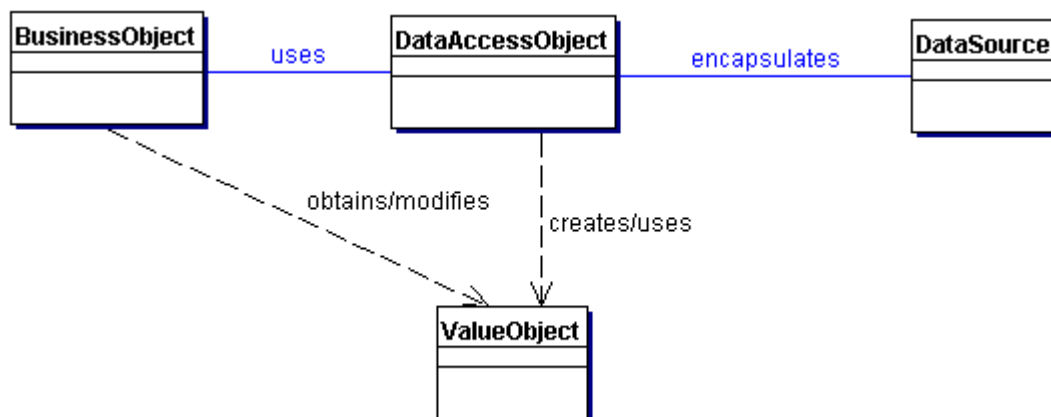


Figura 6: Diagrama representativo del patrón DAO

Respecto de la utilización del patrón en el proyecto de acuerdo a su usabilidad, indicar que normalmente se crea un DAO por cada Objeto que se usa en la aplicación. A modo de ejemplo, un objeto de este tipo sería el marco o *framework* de persistencia, representando el acceso a cada una de las tablas de nuestro sistema de base de datos.

El esquema mostrado en la *figura 7*, se refiere a la distribución de DAO:

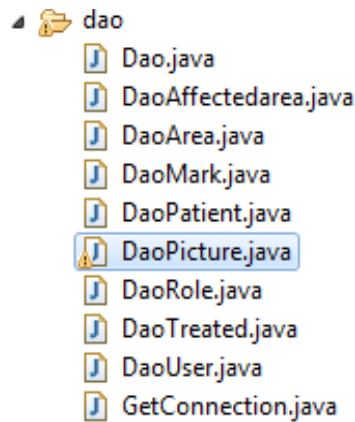


Figura 7: Diagrama representativo del patrón DAO

Patrón Factoría abstracta

En diseño de software, el patrón de diseño Factory Method consiste en utilizar una clase constructora interfaz conteniendo una serie de métodos definidos, tal y como se muestra en el siguiente ejemplo:

```
package dao;

import java.util.ArrayList;

public interface Dao {
    ArrayList<Object> all = null;

    public void create(Object object) throws Exception;

    public Object get(Object object) throws Exception;

    public boolean delete(Object object) throws Exception;

    public boolean update(Object object) throws Exception;

    public ArrayList<Object> getAll() throws Exception;
}
```

Esta interfaz únicamente expresa qué acciones se van a realizar, pero no como se realizarán. Por consiguiente, una clase que herede de la interfaz se obliga a implementar el cómo realizará la implementación de dichos métodos o acciones.

```
public class DaoPicture implements Dao {  
  
}
```

Por consiguiente, en relación a la usabilidad del patrón en el proyecto y según lo mencionado previamente, se consigue homogeneizar el diseño y estructura del proyecto ante futuras mejoras.

3.4.2 Plataforma y requerimientos derivados para el desarrollo de la aplicación

En cuanto a la plataforma para el desarrollo de la aplicación DermaSystem, indicar que se ha empleado eclipse Juno en su versión Juno Service Release 2 y como máquina virtual java JVM v1.8 con Java ToolKit v8.

Los sistemas operativos en los que se puede continuar el desarrollo y sus requisitos son los siguientes:

Windows

- Windows 8 (escritorio)
- Windows 7
- Windows Vista SP2
- RAM: 128 MB; 64 MB para Windows XP (32 bits)
- Espacio en disco: 124 MB

Mac OS X

- Mac basado en Intel que ejecuta Mac OS X 10.7.3 (Lion) o posterior.
- Privilegios de administrador para la instalación

Linux

- Oracle Linux 5.5+
- Oracle Linux 6.x (32 bits)*, 6.x (64 bits)*

- Red Hat Enterprise Linux 5.5+, 6.x (32 bits)*, 6.x (64 bits)*
- Ubuntu Linux 10.04 y superior
- SUSE Linux Enterprise Server 10 SP2, 11.x

3.4.3. Base de Datos

Premisas de partida del proyecto

Al tratarse de un proyecto heredado, un importante esfuerzo se ha centrado en ampliar el diseño de la base de datos para adaptarla a las nuevas necesidades del mismo.

En este sentido, conviene tener en cuenta que este proyecto no está subvencionado por ningún cliente por lo que existen mejores tecnologías comerciales que las mencionadas previamente, la elección de las utilizadas en el presente proyecto no implican ningún coste adicional y además cumplen perfectamente los requerimientos exigidos al proyecto, ya que en esencia es un prototipo o germen de un proyecto más amplio y ambicioso.

Sistema gestor de base de datos o SGBD y hospedaje

MySQL es el sistema de gestión de base de datos utilizado. Es un sistema de gestión de bases de datos relacionales que se ejecuta como un servidor, proporcionando acceso de múltiples usuarios a una serie de bases de datos. Ha sido elegido por ser uno de los más utilizados por la comunidad internacional en proyectos de similar naturaleza al aquí desarrollado y por ser de naturaleza *open source* (código abierto). Al trabajar sólo con unos cientos de datos, MySQL se amolda perfectamente a las necesidades de DermaSystem.

El servidor de base de datos se aloja de forma online por la facilidad que ofrece a la hora de centralizar la base de datos en un servidor en la nube, lo que permite su acceso a la hora de trabajar en ordenadores diferentes. El sitio elegido con tal finalidad ha sido www.db4free.net que nos permite realizar tal hospedaje sin ningún tipo de coste económico. La velocidad de acceso a datos de este servidor no es muy alta, si bien esto no representa ningún problema para el correcto funcionamiento de la aplicación.

Modelo entidad relación

Para realizar el modelo relacional se ha empleado la herramienta de diseño MySQLWorkBeanch la cual permite plasmar los cambios que se desean realizar en la base de datos de una manera gráfica, a la vez que obtener el script para su despliegue

en el repositorio de web de la misma con el fin de poder ser compartida por todos los usuarios de la aplicación y los miembros del equipo en su desarrollo.

En la *figura 8* se muestra el diagrama entidad-relación correspondiente a la base de datos diseñada. La parte enmarcada en color rojo corresponde al diseño de las nuevas tablas añadidas en la ampliación del proyecto a partir de la versión original de DermaSystem (Llorente-Navarro y col., 2013).

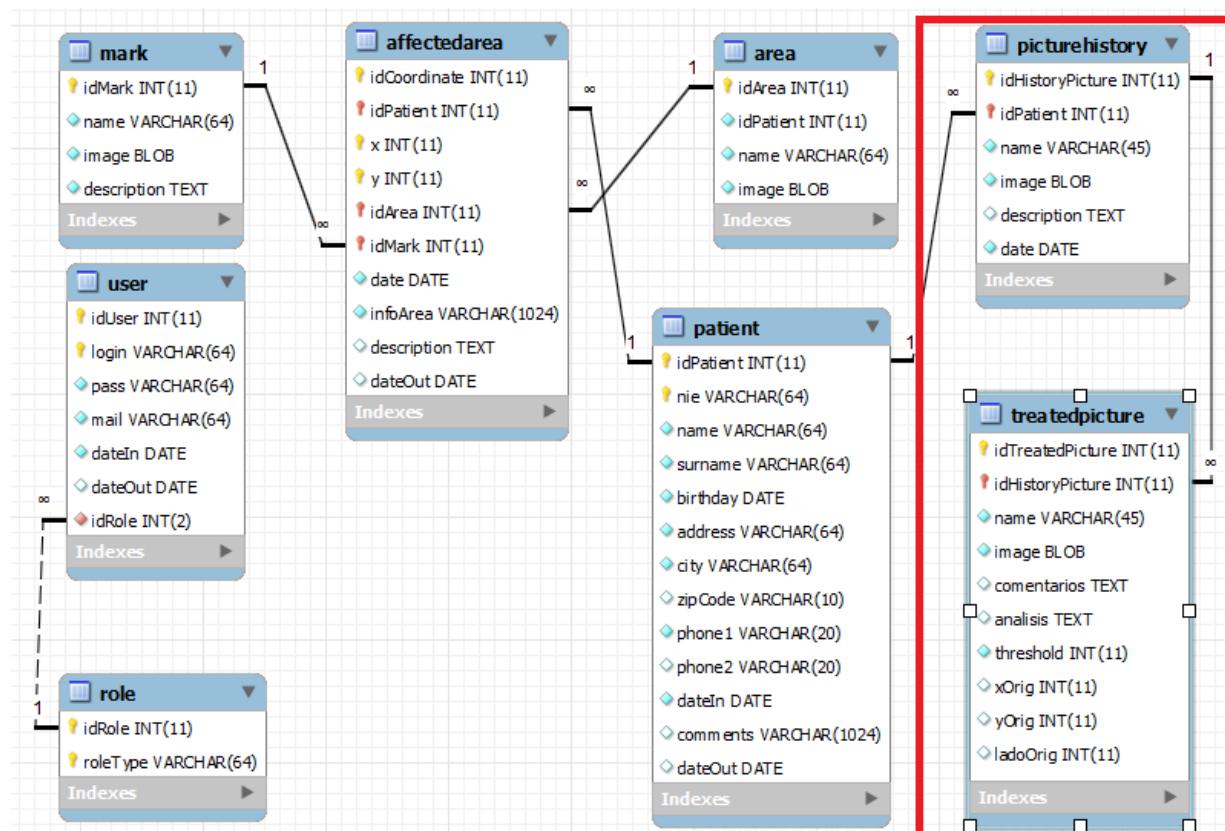


Figura 8: Diagrama entidad-relación de la base de datos

Detalles de las nuevas tablas añadidas se concretan como sigue:

- Tabla *picturehistory*: esta tabla contiene las imágenes originales de los pacientes con toda la información relevante, tal como, nombre, fecha de carga de la imagen en el sistema y una descripción de la misma sobre los motivos por los que ha sido considerada de interés por el médico dermatólogo.
- Tabla *treatedPicture*: es el lugar donde persisten los datos necesarios para recrear la aplicación de los métodos de tratamiento de las imágenes originales. También se guardan comentarios relacionados con los diagnósticos resultantes tras la aplicación de los citados métodos por el usuario médico.

3.5. Diagrama de Clases

La *figura 10* muestra el correspondiente diagrama de clases de clases sobre el que se fundamentaba la primera versión de DermaSystem.

En la *figura 9* se muestra el diagrama de clases en detalle que muestra la expansión del anterior, como consecuencia de la incorporación de las nuevas funcionalidades incorporadas en la nueva versión de DermaSystem, tal y como se recogen en la presente memoria.

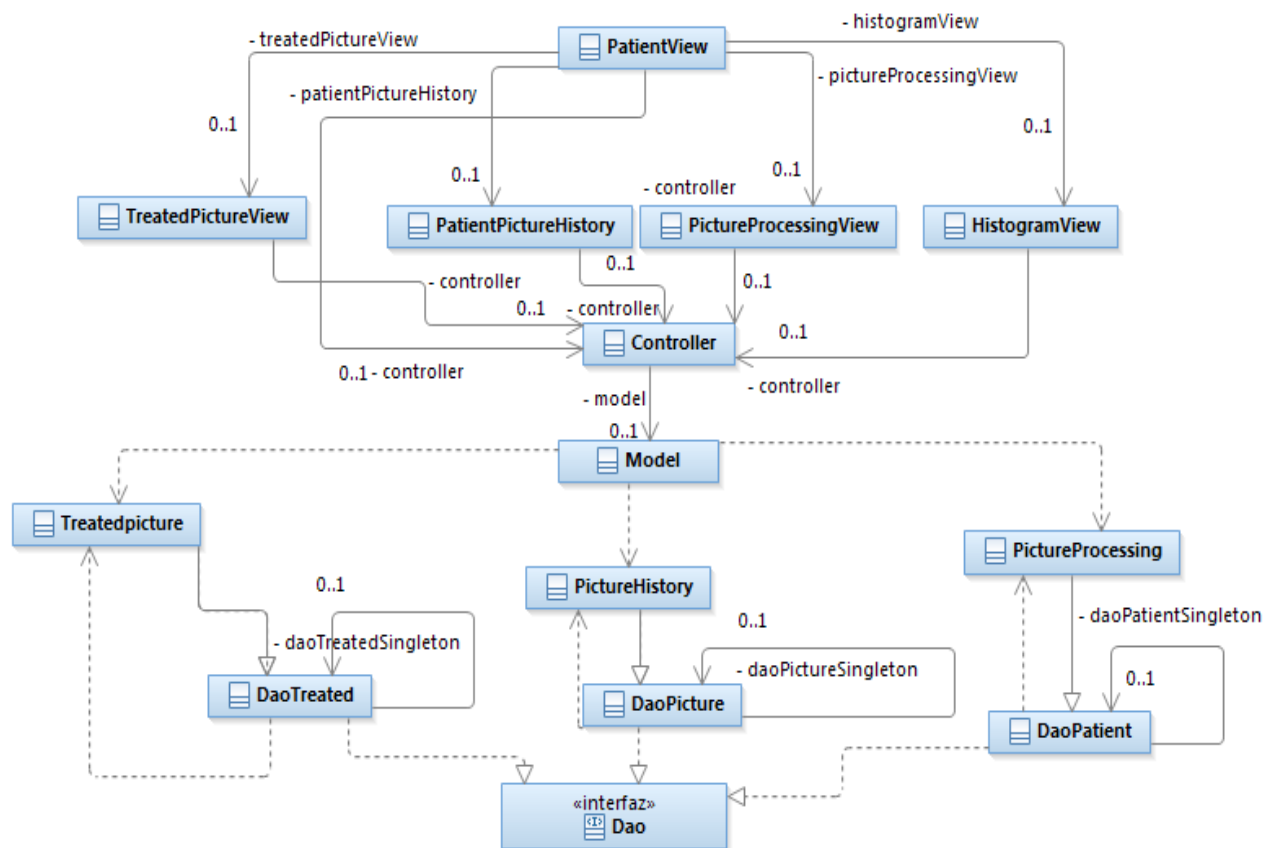


Figura 9: Diagrama de clases detallado relativo a la ampliación de DermaSystem

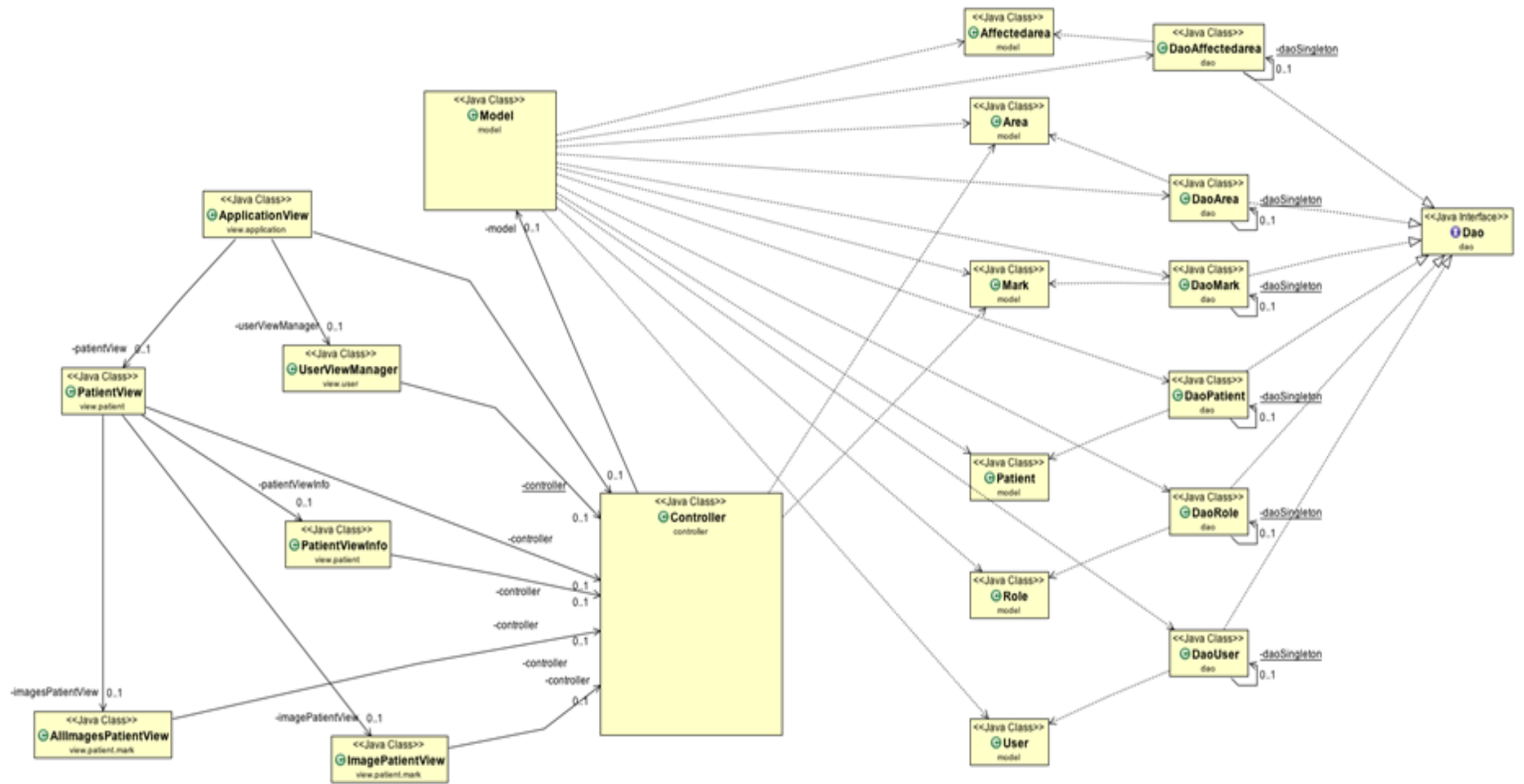


Figura 10: Diagrama de clases del proyecto DermaSystem en su primera versión

3.6. Diagramas de secuencia

Como bien es sabido, un diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el dicho escenario y mensajes intercambiados entre los objetos.

Por este motivo hemos seleccionado dos operaciones que conllevan el mayor número de interacciones entre objetos posibles para poder obtener una mejor comprensión global de la aplicación.

3.6.1. Diagrama de secuencia subir imagen al historial del paciente

El diagrama de secuencia de la *figura 11* muestra el esquema de diseño para la secuencia subir imagen al historial del paciente. La descripción de las acciones es la que se muestra a continuación:

- El usuario desde la perspectiva “historial de imágenes” solicita “cargar nueva imagen”.
- Desde el diálogo que se muestra tras la acción anterior, el usuario irá navegando por el árbol de directorios hasta encontrar la imagen que desea añadir al historial del paciente. La imagen tiene que estar en formato JPEG, elegido por su extendido uso en multitud de dispositivos y aplicaciones.
- La vista realiza la invocación al controlador con los datos recibidos de la ventana emergente y éste a su vez los transmite al modelo.
- El modelo llama al DAO, éste crea una conexión con la base de datos y ejecuta una instrucción del tipo “INSERT” en la base de datos añadiendo una nueva entrada en la misma.
- Una vez persistidos los datos se realiza una actualización del historial de imágenes de las que dispone dicho paciente (Clase log).
- El modelo lanza un evento indicando que se ha realizado correctamente la creación, el cual es capturado por la vista, que a su vez presenta finalmente una notificación de que dicha operación se ha realizado satisfactoriamente.

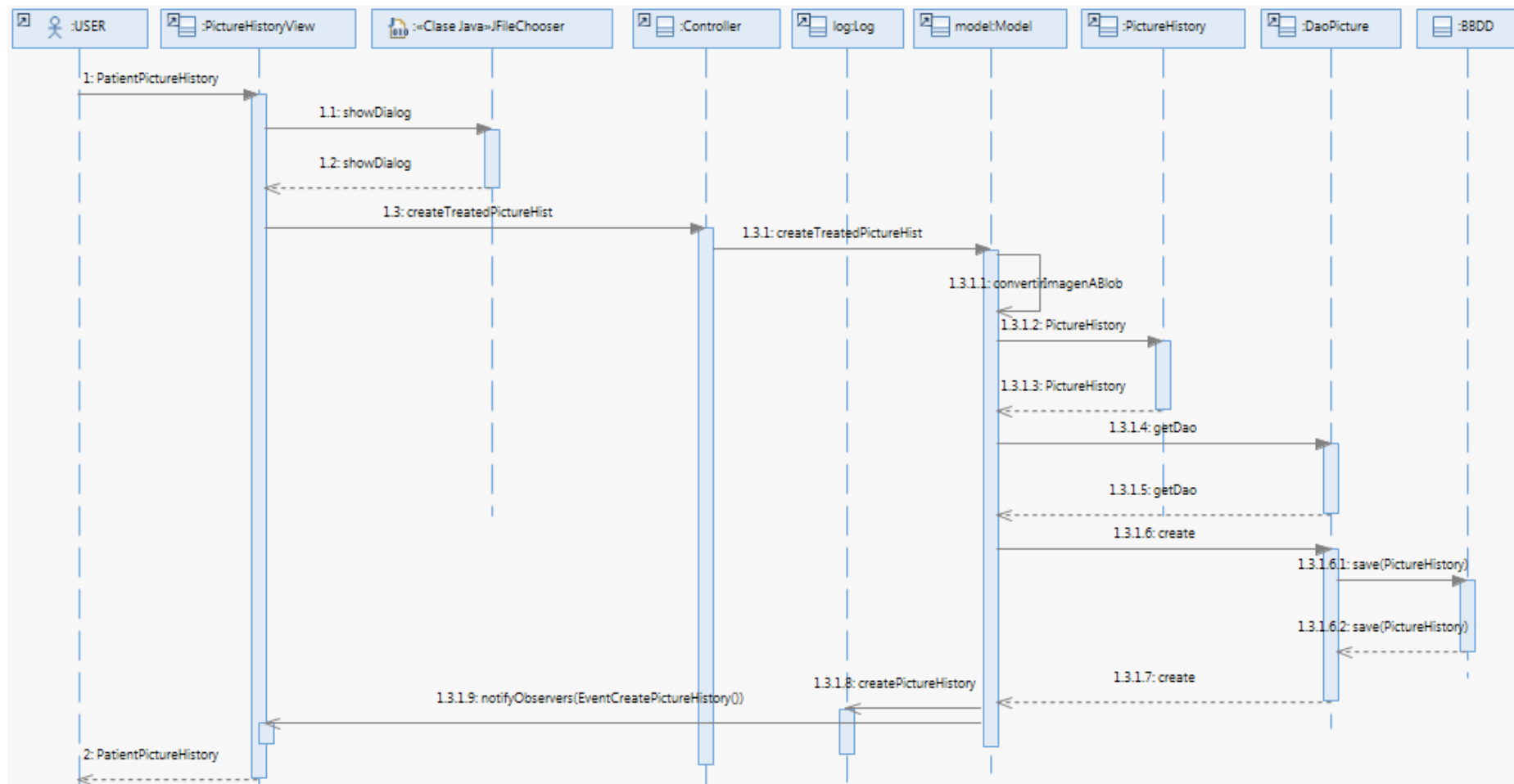


Figura 11: Diagrama de secuencia “subir imagen al historial del paciente”

3.6.2 Diagrama de secuencia de analizar imagen

En la *figura 12* se muestra el diagrama de secuencia analizar imagen. La descripción de las acciones es la que se muestra a continuación:

- El usuario desde la perspectiva de “historial de imágenes” selecciona la imagen original que desea tratar.
- Se le aplica un proceso de binarización a la imagen original seleccionada a través de las funciones y algoritmos proporcionados por las librerías de tratamiento de imágenes. Devolviéndole dicha imagen al usuario.
- El usuario selecciona la operación que desea realizar a través de la correspondiente función sobre una imagen, invocándose a la función umbral o *threshold* con los parámetros específicos de dicho filtro, tras lo cual se pasa el control desde la vista al controlador.
- El controlador llama a la función del modelo la cual devuelve la imagen aplicándole la operación seleccionada para posteriormente pasarlo a la vista.
- El usuario desencadena la llamada “createTreatedPicture” desde la vista, la cual le pasa la llamada al controlador.
- El controlador verifica los datos llamando al modelo y éste a través del *daoTreated* ejecuta una acción del tipo “save” en la base de datos, asignándole un nuevo identificador.
- Se notifica a la vista que todo ha ido correctamente, ésta a su vez muestra un mensaje confirmando el éxito de la operación.

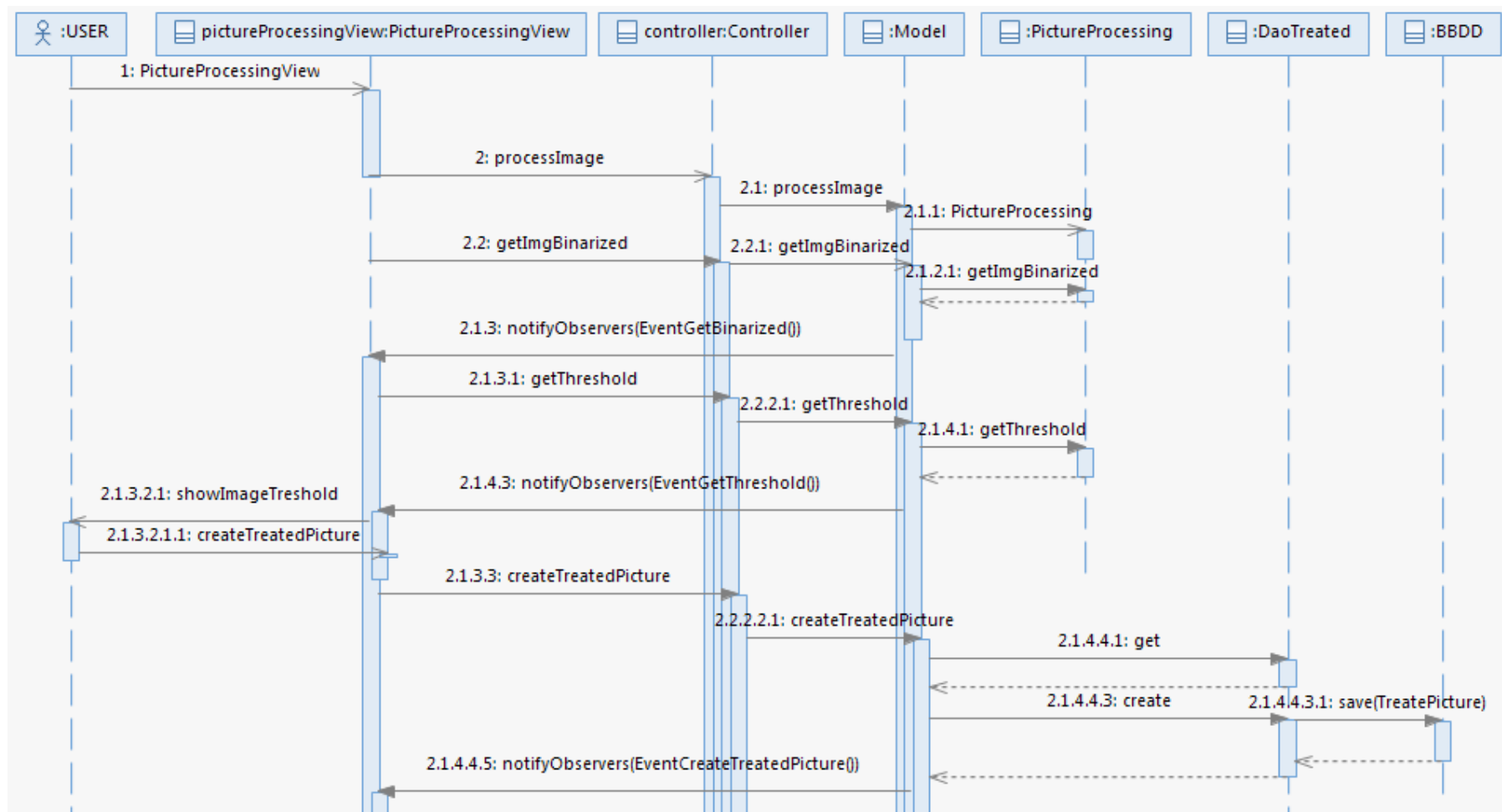


Figura 12: Diagrama de secuencia “analizar imagen”

4. Implementación

Una vez se dispone del diseño de la aplicación, el paso siguiente es su implementación. En este capítulo se describen por un lado las tecnologías utilizadas con tal propósito, así como los interfaces previstos con tal finalidad.

4.1. Tecnologías utilizadas

En esta sección se enumeran y detallan cada una de las tecnologías utilizadas en la implementación de DermaSystem en su versión actual, cuyo diseño se corresponde básicamente con el realizado previamente, lo que implica la coincidencia de la mayoría de las tecnologías utilizadas en la versión original descrita en Llorente-Navarro y col., (2013).

4.1.1. Java

DermaSystem ha sido programado en Java en su totalidad, debido a que, al tratarse de una aplicación con gran carga de interacción con el usuario, se ha querido utilizar un lenguaje que facilitase la programación dirigida por eventos.

Gracias a las prestaciones que ofrece este “tipo” de programación, se han logrado controlar de manera eficiente las actividades que llevan a cabo tanto usuario como aplicación.

4.1.2. Java Swing

Java Swing es una herramienta de diseño de interfaces gráficas de usuario, conocido usualmente como GUI, del inglés *graphical user interface*. Su popularidad y su gran cantidad de componentes, lo convierten en la herramienta ideal para desarrollar la interfaz de DermaSystem, puesto que permite crear la experiencia de usuario deseada de una manera fácil y eficaz.

4.1.3. Java Advanced Imaging

Java Advanced Imaging, o como se conoce habitualmente, JAI (2013), es una extensión de Java que proporciona un conjunto de interfaces orientadas a objetos, permitiendo a los desarrolladores crear sus propias rutinas de manipulación de

imágenes, sin coste adicional o restricciones de licencias relacionadas con los procesadores de imágenes comerciales. Esto convierte a JAI en la herramienta ideal para la implementación del módulo de tratamiento de imágenes que forma parte de DermaSystem.

4.1.4. JCalendar

JCalendar es una librería de Java que permite seleccionar una fecha de manera gráfica y sencilla, mostrando un calendario.

Está compuesta por una serie de elementos tales como JDayChooser, JMonthChooser y JYearChooser, los cuales poseen métodos que permiten ser utilizados fácilmente en constructores GUI.

4.1.5. JFileChooser

La clase java JFileChooser proporciona un interfaz de usuario para elegir un fichero de una lista disponible de ficheros. Un selector de ficheros (filechooser) es un componente que podemos situar en cualquier lugar del GUI del programa que implementa la aplicación DermaSystem. La clase JFileChooser hace sencillo traer un diálogo modal que contiene un selector de ficheros, permitiéndonos la importación del mismo al sistema desarrollado.

4.1.6. Hibernate

Hibernate (2014) es una herramienta de Mapeo objeto-relacional, conocida como ORM, para la plataforma Java, que facilita el mapeo de atributos entre una Base de datos relacional tradicional y el modelo de objetos de una aplicación.

La utilización de Hibernate permite solucionar el problema existente entre los datos con los que trabaja la aplicación DermaSystem, y los datos que posteriormente se guardarán en la base de datos.

Además, el uso de la base de datos es totalmente transparente, pudiendo cambiar dicha base sin necesidad de modificar código en la aplicación, tan sólo se modificarían los ficheros de configuración de Hibernate y se añadirían las consultas deseadas en las clases DAO. Esta cualidad resulta ser muy práctica para ampliar las funcionalidades de DermaSystem en el futuro.

El funcionamiento de hibernate se sintetiza como sigue:

- Hibernate asocia a cada tabla de la base de datos un Plain Old Java Object, más conocido como POJO. Un POJO es similar a una clase, con propiedades accesibles mediante métodos setter y getter.
- Para poder asociar el POJO a su tabla correspondiente en la Base de datos, Hibernate usa archivos XML, en los que se declaran las propiedades del POJO y sus correspondientes nombres de columna para cada tabla en la base de datos. También se puede incluir la asociación de tipos de datos, referencias, relaciones del tipo x a x con otras tablas, etc.

4.1.7. *Asamblea*

Asamblea (2014) es una empresa que provee herramientas de colaboración y de seguimiento de errores (Bug Tracking System) y tareas basadas en la nube para organizar y administrar proyectos de código abierto y comerciales para el desarrollo de software. En concreto, para la realización de este proyecto se ha utilizado su producto Espacios de Trabajo (Project Workspaces), el cual ofrece un control de versiones o Subversion (2014).

Subversion

Subversion (2014) es una herramienta de control de versiones de tipo *open source* basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras ubicadas en diferentes lugares geográficos. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta y facilita la colaboración, lo cual resulta de especial interés en colaboraciones entre profesionales en dermatología.

4.1.8. *IDEs*

Para la implementación de DermaSystem, como entorno de desarrollo integrado, se ha elegido utilizar Eclipse, el cual es un programa de código abierto que proporciona una inmensa funcionalidad gracias a extensiones, en inglés *plug-ins*. Además de ser un programa muy completo, en cuanto a prestaciones se refiere, se ha decidido optar por Eclipse al ser el entorno de desarrollo en el cual han efectuado su actividad estudiantil los participantes en el proyecto durante sus años de formación en la Facultad de Informática.

4.1.9. Subclipse

Subclipse es un plug-in para Eclipse, el cual proporciona soporte a Subversion dentro del IDE de Eclipse. Esto ha permitido que cada miembro del grupo pudiera trabajar en el desarrollo del proyecto por separado, aprovechando el repositorio de Subversion para sincronizar el trabajo individual de cada uno.

4.1.10. Trello

Trello es una aplicación web que permite la organización de proyectos de una manera sencilla. Utiliza un método denominado kanban, el cual se basa en la representación de los proyectos como tablas que contienen tres listas de tareas: pendientes, en progreso y completas. Las listas contienen tarjetas, cada una de ellas representa una tarea y uno o varios usuarios que la deben realizar. Las tarjetas deben pasar de una lista a la siguiente según el estado en el que se encuentren.

4.2. Interfaces

Esta sección describe los interfaces de la aplicación, haciendo hincapié en la parte novedosa respecto de la sección equivalente en la descripción del sistema en su versión inicial descrita en Llorente-Navarro y col., (2013). Las diferentes vistas que incluyen los interfaces son los que se describen a continuación.

4.2.1. *PatientPictureHistory*

Parte fundamental de la aplicación, *PatientPictureHistory* es un `JInternalFrame` especialmente diseñado para mostrar una galería de imágenes, así como la vista de la imagen seleccionada en la misma. Desde esta vista, se proporcionan distintos botones de herramientas y navegación:

- Borrar
- Analizar
- Volver a marcas
- Cargar imagen
- Detalles
- Imágenes relacionadas

Es un módulo que posee una alta carga de transferencia de datos con la base de datos, dado que no sólo recupera las imágenes desde la misma, sino que también debe guardar las posibles imágenes que pueden escogerse mediante un selector de archivos. De entre todas las opciones disponibles, son dos los botones que poseen una especial relevancia ya que profundizan la ejecución de la aplicación:

- *Cargar imagen*: permite al usuario añadir imágenes al historial de un paciente para su posterior tratamiento.
- *Imágenes relacionadas*: añade un nivel más al sistema, puesto que con él se abre una nueva vista, llamada *TreatedPictureView*, que se explicará más adelante.
- *Analizar*: botón fundamental para el propósito del proyecto, abre la vista *PictureProcessingView* para analizar la imagen que está siendo visualizada.

4.2.2. *PictureProcessingView*

En este caso se trata de una de las vistas fundamentales para cumplir los nuevos requisitos del proyecto, proporciona al usuario las herramientas necesarias para realizar el análisis de la imagen.

PictureProcessingView es un *JInternalFrame* cuya función principal es mostrar la imagen seleccionada previamente, tanto en su forma original como después del tratamiento. Ofrece herramientas cuyas finalidades son las que se especifican a continuación:

- Recortar la imagen
- Guardar la imagen
- Restaurar la imagen original
- Umbralizar
- Seleccionar el valor de umbral: mediante un valor numérico o con deslizador
- Automatizar el valor de umbral
- Extraer información de la imagen
- Mostrar el histograma

Para completar la funcionalidad de las dos últimas son necesarias las dos vistas flotantes nuevas:

- *ImageTable*: esta vista muestra de la forma más clara y ordenada posible la información extraída del análisis de la imagen. En una tabla, inserta los valores medios de intensidad de los tres canales de color R, G y B correspondientes a los valores de los píxeles de la región marcada en negro tras la umbralización, así como sus varianzas, la energía y la entropía, esto es los valores estadísticos, como descriptores de textura de la región a analizar. En la parte inferior, se muestra el valor en píxeles del área de dicha región.
- *HistogramView*: el objetivo de esta vista es simple, mostrar un gráfico de manera sencilla del histograma de la imagen, donde se puedan apreciar picos y desviaciones que ayuden en la apreciación y diagnóstico de la mancha o lesión en concreto. Además, gracias al paquete *jmathplot* (2014), se proporcionan herramientas suficientes para aumentar el nivel de zoom y escala de los ejes de coordenadas en la ventana de representación de la imagen.

4.2.3. *TreatedPictureView*

Esta vista es en realidad un *JInternalFrame* que proviene de una llamada de *PatientPictureHistory*, ocupando su lugar para mostrar una imagen que ha sido tratada previamente y está guardada en el historial. Para interactuar con ella, *TreatedPictureView* proporciona al usuario médico los siguientes botones asociados a las correspondientes funcionalidades.

- Volver
- Eliminar
- Detalles
- Navegación en la galería

En su conjunto proporcionan las acciones necesarias para visualizar todos los tratamientos previos que se han realizado sobre la imagen seleccionada. De esta forma es posible comparar las distintas posibilidades de análisis para decidir cuál es la más apropiada de cara a la ayuda y soporte de la aplicación que se pretende.

4.2.4. *PatientViewInfo*

Esta vista es esencialmente la misma que su homóloga descrita en la memoria DermaSystem (Llorente-Navarro y col., 2013), si bien con la salvedad de que posee un botón nuevo que se ha añadido para ampliar la información del paciente.

El botón “Historial de imágenes” habilita la nueva vista *PatientPictureHistory* comentada anteriormente, que constituye el punto de comienzo del flujo principal relativo a la ampliación del proyecto DermaSystem a partir de su versión original.

5. Conclusiones y futuro

5.1. Conclusiones

DermaSystem es una herramienta informática, diseñada con el propósito de servir de apoyo en las consultas médicas en el área de dermatología. Se desarrolla con una doble finalidad:

- Como ayuda en la gestión de pacientes mediante el almacenamiento de datos e imágenes, facilitando el acceso a los mismos de forma gráfica y textual, con la posibilidad de generación de informes.
- Como ayuda al diagnóstico, mediante el diseño de un módulo de procesamiento de imágenes con el que es posible obtener información sobre diversos parámetros de las lesiones y manchas del paciente (color, textura, tamaño).

El sistema en su conjunto se ha diseñado en base a una metodología y estructura apropiadas para obtener una aplicación robusta y útil, cuyos módulos principales, base de datos y tratamiento de imágenes, cumplen con los objetivos previos. Por otra parte, al tratarse de una aplicación de uso diario en las consultas, se ha diseñado una interfaz gráfica en la que se puede acceder a todas las funcionalidades del sistema de una manera rápida y sencilla.

La aplicación DermaSystem maneja información privada de pacientes, por lo que se ha incluido un sistema de autenticación de usuarios tal que se pueda garantizar la seguridad de los datos. Además, se ha separado la administración del sistema de la aplicación relativa a las funcionalidades que utilizará el dermatólogo.

La aplicación, inspirada inicialmente en las ideas y necesidades de la doctora D^a María Isabel Longo Imedio, se ha diseñado e implementado con previsión de futuro para posibles ampliaciones. En resumen, en este momento se dispone de una aplicación que permitirá manejar los historiales clínicos de los pacientes, con un sistema de marcado de las afecciones dermatológicas, que facilitará el control sobre las mismas. Además, la aplicación contiene un módulo de tratamiento de imágenes, con el cual el médico obtiene una información detallada sobre la lesión reflejada en la imagen analizada.

La parte de gestión de la base de datos se encuentra bastante avanzada en el momento actual. Respecto del módulo de tratamiento de imágenes, se ha realizado un primer desarrollo, no incluido en la versión original de DermSystem. Gracias a ello se ha podido comprobar la posibilidad de realizar tratamientos de imágenes, almacenando la información extraída, compatible con el resto de funcionalidades de gestión. Es hora de ahondar en este sentido en futuras versiones del sistema, dotándole de funcionalidades de mayor calado orientadas hacia técnicas de aprendizaje e inteligencia artificial en el ámbito de la visión por computador.

5.2. Futuro de la aplicación

DermaSystem es un proyecto con una enorme proyección de futuro, que hasta el momento, cubre las necesidades básicas que puede tener un especialista en dermatología durante la consulta. Desde el comienzo, la aplicación se ha diseñado para su continuidad de futuro, con idea de poder ser ampliada con facilidad, de manera que se pueda añadir funcionalidad sin la necesidad de iniciar el proyecto desde cero o estudiar a fondo la implementación anterior.

A continuación se indican una serie de funcionalidades y aspectos que en el futuro se podrían añadir a DermaSystem como continuación y mejora del mismo:

- Marcado automático de imágenes tras su procesamiento y análisis de su evolución.
- Posibilidad de añadir nuevas marcas gráficas relativas a diversas lesiones.
- Elaboración gráfica de informes sobre la información contenida en la base de datos.
- Impresión de los informes conteniendo todas las áreas con sus correspondientes marcas e imágenes, tanto originales como tratadas.
- Añadir nuevos tratamientos a las imágenes, con el fin de obtener la máxima información posible sobre la lesión o mancha reflejada en la imagen.
- Añadir funcionalidades de procesamiento automático de imágenes, basadas en métodos de aprendizaje, como parte esencial de ayuda al diagnóstico.

Todos estos puntos, junto con las funcionalidades y aspectos ya desarrollados e implementados en la aplicación, suponen un producto final de DermaSystem que sin duda facilita la gestión informática de pacientes y en el futuro servirá como ayuda base al diagnóstico médico en dermatología.

Conclusions and future work

DermaSystem is a computer tool designed with the purpose of supporting medical consults in dermatology area. It has been developed with two aims:

- As a help in patient management by means of data and images storage, making easy access easy in a graphical and textual way, with the possibility of informs generation.
- As a help to diagnostic, thanks to an image processing module which makes it possible to obtain information about several parameters in skin injuries, rashes or spots of the patient (color, texture or size).

The system has been designed based on proper methodology and structure to obtain a consistent and useful application, with two new modules that cover the previous objectives. As a daily use application for consults, it has been designed a graphic interface in which the user can access all the functionalities easily and quickly.

In addition, DermaSystem manages patients private information, so it has been included a user authentication system that guarantees data security. Thus, system administration has been separated from the application part related to the functionalities that the dermatologist would use. Besides, this software, initially designed by the ideas and necessities of Medicine doctor Ms. Isabel Longo Imedio, has been designed and implemented with the possibility of extension in the future.

To sum up, in this moment this application permits to manage patients' medical histories, with a system to mark the dermatologic illnesses that will provide control over them, as well as an image processing module that the doctor can use to obtain detailed information about the injury seen in the analyzed image.

The database management part is quite advanced at this moment. On the other hand, the image processing module is in its first revision, because it was not included in the previous version of DermaSystem.

The following is a list of functionalities that could be added in the future to DermaSystem:

- Automatic labelling of the images, once they have been processed.
- Addition of new graphic labels related to injuries.
- Graphical reports generation about the information stored in the Database.
- Printing of reports with all areas and preserving the annotated labels.
- Add new image processing techniques, to extract all information from the images.

- Include learning-based methods to design an intelligent system that learns with the information processed.

All these points, together with all functionalities included in the current version of DermaSystem, make it a product of great help in dermatology.

6. Glosario

B2B: Business-to-Business

BD: Base de Datos

BSD: Berkeley Software Distribution

CORBA: Common Object Request Broker Architecture

DAO: Data Access Object

DB: Data Base

GUI: Graphical User Interface.

IDE: Integrated Development Environment

IOP: Internet InterORB Protocol

IP: Image Processing

JAI: Java Advanced Imaging

JPEG: Joint Photographic Experts Group

JVM: Java Virtual Machine

LDAP: Lightweight Directory Access Protocol

MD5: Message-Digest algorithm 5

MVC: Modelo Vista Controlador

ORM: Object-Relational Mapping

POJO: Plain Old Java Object.

RAM: Random Access Memory

RDBMS: Relational DataBase Management System

SGBD: Sistema de Gestión de Bases de Datos

SQL: Structured Query Language

TI: Tratamiento de Imágenes

XML: eXtensible Markup Language.

7. Bibliografía

1. *Assembla*. (2014). <https://www.assembla.com/home>
2. Bloch J. (2008). *Effective Java*. 2ª Ed. Addison-Wesley.
3. Gamma E., Helm R., Johnson R., Vlissides J. (2005). *Design Patterns: Elements of Reusable ObjectOriented Software*. Edición Internacional. CBR.
4. *Hibernate*. (2014). <http://hibernate.org/>
5. *JAI Java Advanced Imaging API* (2014). Oracle.
<http://www.oracle.com/technetwork/java/javase/tech/jai-142803.html>
6. *JMathPlot* (2014). Interactive 2D and 3D plots.
<http://jmathtools.berlios.de/doku.php>
7. Llorente-Navarro, S.; Muñoz-Díaz, C.M.; Olivar-Conde, J.C. (2013). *DermaSystem: Gestión Informática en Dermatología*. Proyecto Fin de Grado. Facultad de Informática. Universidad complutense de Madrid. Madrid.
8. *MATLAB. The Mathworks* (2014). <http://www.mathworks.es/>
9. Otsu, N. (1979). *A threshold selection method from gray-level histogram*, *IEEE Transactions on System Man Cybernetics*, Vol. SMC-9, No. 1, 1979, pp. 62-66.
10. Pajares G., de la Cruz J.M. (2007). *Visión por computador: imágenes digitales y aplicaciones*. 2ª Ed. RA-MA.
11. Pajares G., Santos, M. (2005). *Inteligencia Artificial e Ingeniería del Conocimiento*. RA-MA.
12. Pressman, Roger S. *"Ingeniería de Software. Un enfoque práctico"*. 5ª Ed. Editorial Mac Graw Hill, Madrid, España, 2001.
13. Reenskaug T., Coplien J.O. (2009). *The DCI Architecture: A New Vision of Object-Oriented Programming*. http://www.artima.com/articles/dci_vision.html
14. *Subversion*. (2014). <http://subversion.apache.org/>

Anexo

Manual de usuario

A continuación se describen tres Anexos del manual de usuario, relativos a la parte específica que se ha desarrollado en la ampliación de la versión actual de DermaSystem. Más concretamente, son los relativos a: Historial de imágenes, Analizar imagen de un paciente, Historial imágenes tratadas.

El resto de funcionalidades del programa se encuentran en el Anexo correspondiente a la descripción del manual de usuario de la memoria relativa a la versión previa de DermaSystem (Llorente-Navarro y col., 2013), que se enumeran a continuación:

- Acceso a la aplicación
- Cambio de idioma
- Información y ayuda
- Mi cuenta
- Editar mi cuenta
- Gestión de usuarios
- Gestión de pacientes
- Marcas

A1. Historial de imágenes

Requerimientos iniciales:

- Tener acceso a la aplicación con permisos de usuario médico
- Tener un paciente seleccionado.

Es bien conocido que el acceso al gestor de pacientes con permisos de administrador está restringido.

Existe un paso común para todos los puntos siguientes: desde la ventana del paciente, donde se encuentra la información, pulsar en el botón **“Historial de Imágenes”** (figura A1).

DermaSystem

Pacientes Usuarios Ajustes Ayuda

DNI 50765626A

Nombre Maria Luisa

Apellidos Pérex Loméra

Fecha de Nacimiento 1974-07-04

Fecha de alta 2014-05-14

Dirección Calle de la rosa

Ciudad Madrid

Código postal 28016

Teléfono 1 666999777

Teléfono 2

Editar Historial de Imágenes

Observaciones

Volver

Vista Principal del Paciente

Haga click para seleccionar la zona donde desea registrar una marca

Parte frontal del Paciente

Parte de atrás del Paciente

Figura A1: Historial de imágenes

A1.1. Agregar una imagen al historial de un paciente.

Una vez situados en esta ventana **Historial de Imágenes** pulsar en el botón **“Cargar Imagen”** (figura A2). Se abrirá una ventana emergente que nos permitirá navegar por el árbol de directorios para seleccionar la imagen que deseamos asignarle al paciente.

Nota: el sistema sólo permite la carga de imágenes en formato .jpg por lo que sólo se visualizarán aquellas con dicho formato presentes en las carpetas abiertas por el explorador de ficheros.

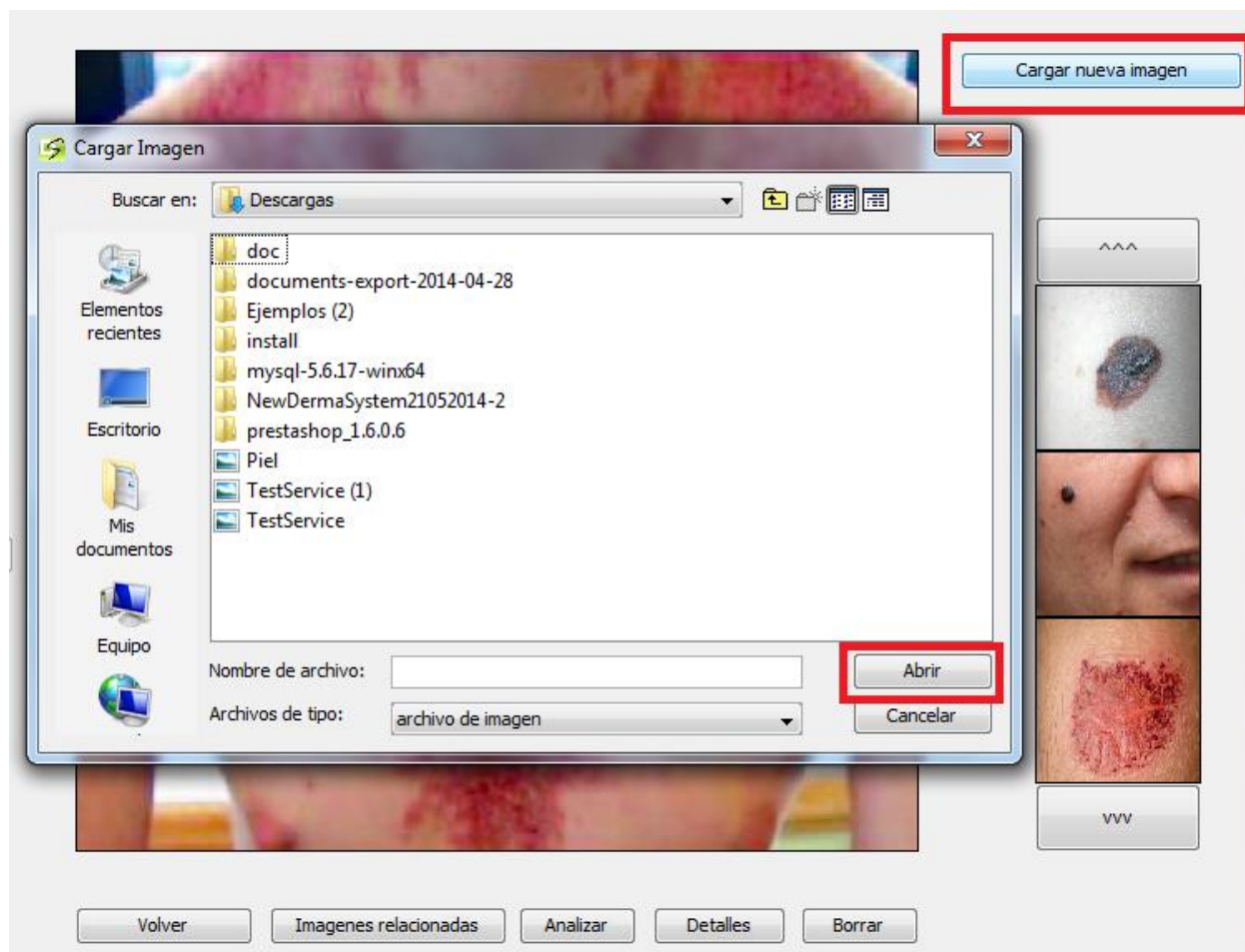


Figura A2: Cargar imagen

Pulsamos “**Abrir**” (*figura A2*) y automáticamente el sistema asigna la referencia de dicha imagen al paciente en el que nos encontramos, al mismo tiempo que refresca la pantalla con la nueva imagen cargada.

A1.2. Borrar una imagen del historial.

Una vez situados en esta ventana **Historial de Imágenes** seleccionamos la imagen que deseamos borrar. Esto implica que dicha imagen tiene que estar ubicada en el lugar central de la pantalla. Una vez asegurados de dicho posicionamiento pulsamos “**Borrar**”.

De esta manera se borrará la imagen y por tanto su vinculación a este paciente. Este paso es irreversible por lo que es necesario considerar esta circunstancia.

A1.3. Consultar detalles de imagen.

Con esta operación el dermatólogo especialista puede consultar los datos y observaciones más relevantes del paciente que hayan sido anotadas o extraídas durante el seguimiento del mismo. Permitiendo así tomar nuevas decisiones sobre el tratamiento.



Figura A3: Detalles de imagen

A1.4. Analizar imagen.

Desde la ventana **Historial de Imágenes** seleccionamos la imagen de la que deseamos consultar su colección de imágenes previas tratadas. Tras esta acción dicha imagen pasa a la posición central de la ventana).

A continuación pulsamos el botón “**Analizar**” (figura A4). Se abrirá una nueva ventana con las posibles operaciones que podemos realizar sobre la imagen original seleccionada en el paso anterior.



Figura A4: Analizar imagen

A1.5. Ver el historial de imágenes tratadas.

De nuevo realizando la misma operación previa desde la ventana **Historial de imágenes** seleccionamos la imagen de la que deseamos consultar su colección de imágenes tratadas.

Finalmente pinchamos en el botón “**Imágenes relacionadas**” (figura A5).

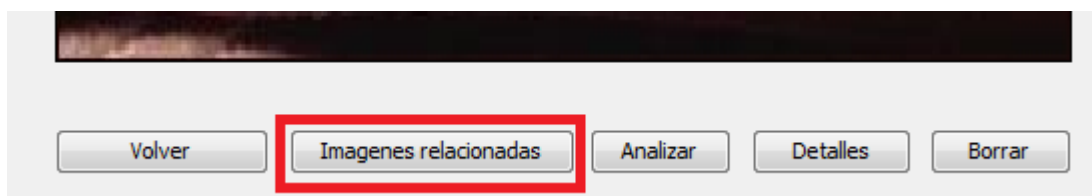


Figura A5: Consultar imágenes relacionadas

Dicha acción nos abrirá una nueva perspectiva donde visualizamos unas imágenes que han sido obtenidas aplicándole el correspondiente tratamiento a la imagen que hemos seleccionado en el paso anterior.



Figura A6: Imágenes relacionadas

A1.6. Volver a la información del paciente.

Una vez situado en la ventana **Historial de imágenes** seleccionamos el botón **“Volver”** (figura A7).

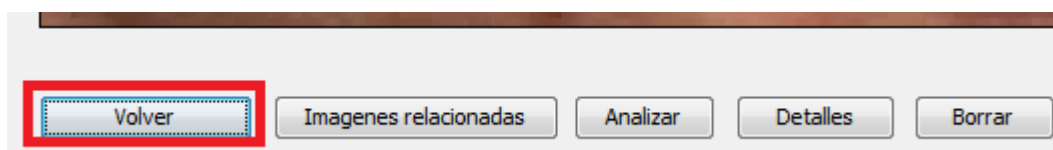


Figura A7: Volver a la información del paciente

El cual nos llevará a la ventana de **Información del paciente** (figura A8) donde se nos muestra su historial de marcas y demás información asociada.

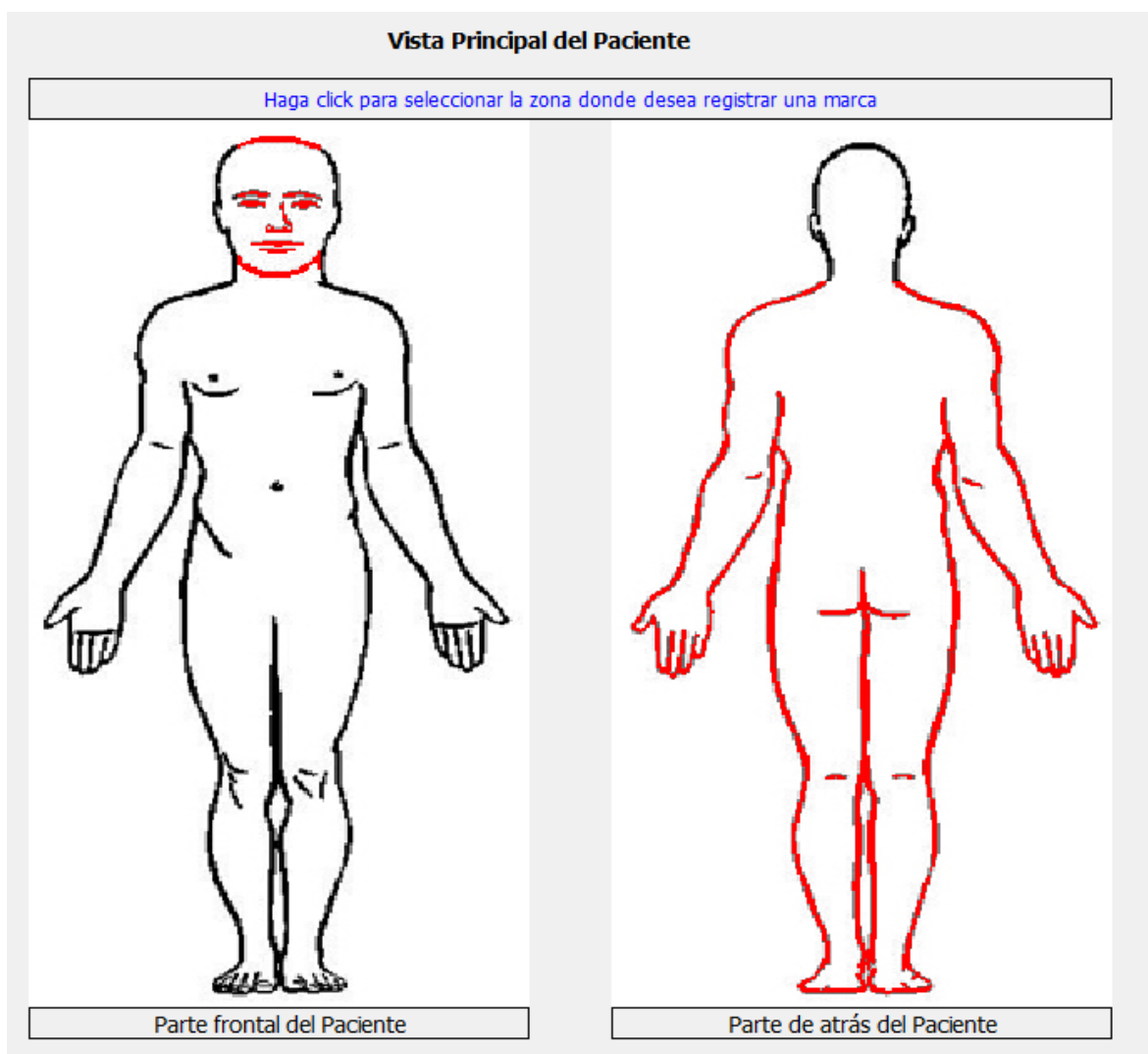


Figura A8: Información del paciente

A2. Analizar imagen de un paciente

Requerimientos iniciales:

- Tener acceso a la aplicación
- Tener un paciente seleccionado
- Acceder a la ventana para análisis de imagen

En este apartado se detallan cada uno de los pasos en el análisis de una imagen. La recomendación para obtener toda la información posible y realizar un análisis completo sigue los siguientes pasos.

- **Recortar** la imagen para ajustarla al tamaño de la lesión.
- **Procesar** la imagen.
- Cambiar el valor de **umbral** para cubrir apropiadamente la zona afectada (parte del procesado de la imagen).
- Obtener la **información** de la imagen.
- Ver el **histograma** de la imagen.
- **Guardar** la imagen para futuras apreciaciones.

Obviamente algunos de los pasos comentados en este flujo de análisis se pueden omitir, por ello se detallan individualmente a continuación.

A2.1. Recortar una imagen

Requerimientos iniciales:

- Situar en la ventana de análisis de imagen.

Situados sobre la ventana del **análisis de imagen**, se puede recortar la misma siempre que se considere necesario, bien para eliminar la parte de piel que no interesa o para ampliar la mancha o lesión, a fin de poder visualizarla correctamente. Para ello, debe hacer clic en “**Recortar**”, representado con el símbolo de las tijeras (*figura A9*):



Figura A9: Tijeras

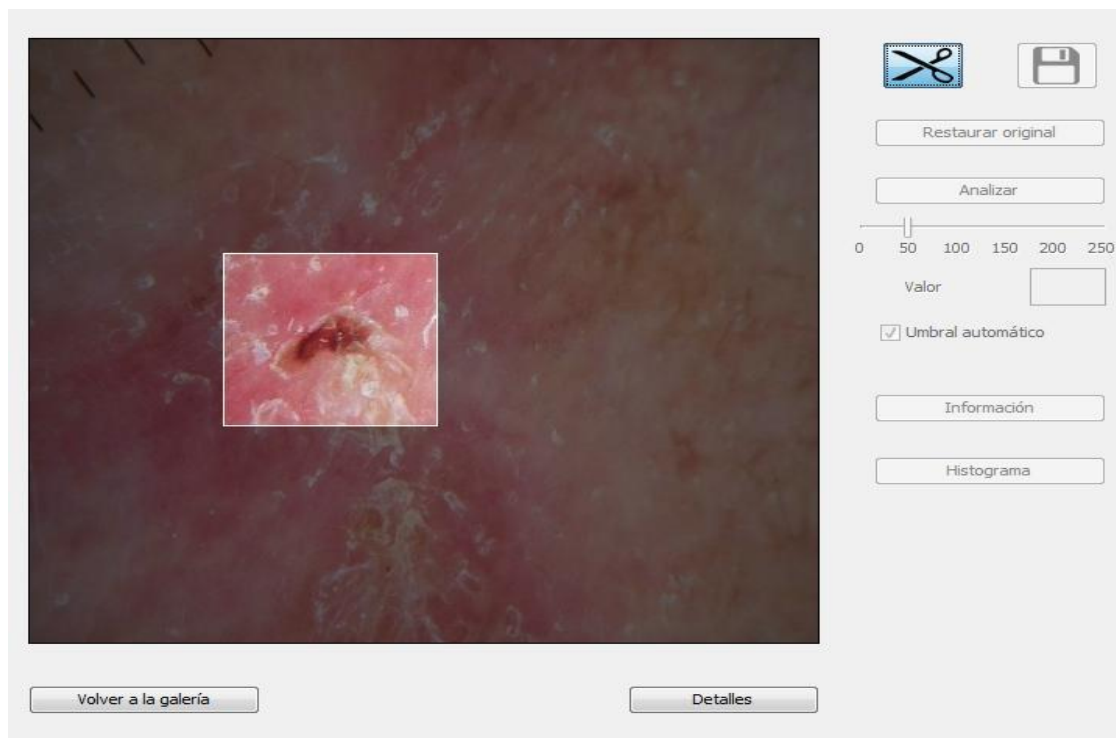


Figura A10: Selección del recorte

Posteriormente, se selecciona un nuevo recorte con el ratón. Al soltar, la imagen se adaptará al ancho de la pantalla. Es necesario tener en cuenta que el recorte va a ser de dimensiones cuadradas antes de comenzar la operación (figura A10).

En caso de necesitar realizar un nuevo recorte, se puede pulsar el botón “**Restaurar original**”. Esto devolverá la imagen a su estado original, sin tratamiento ni recorte, pudiendo comenzar de nuevo la operación (figura A11).

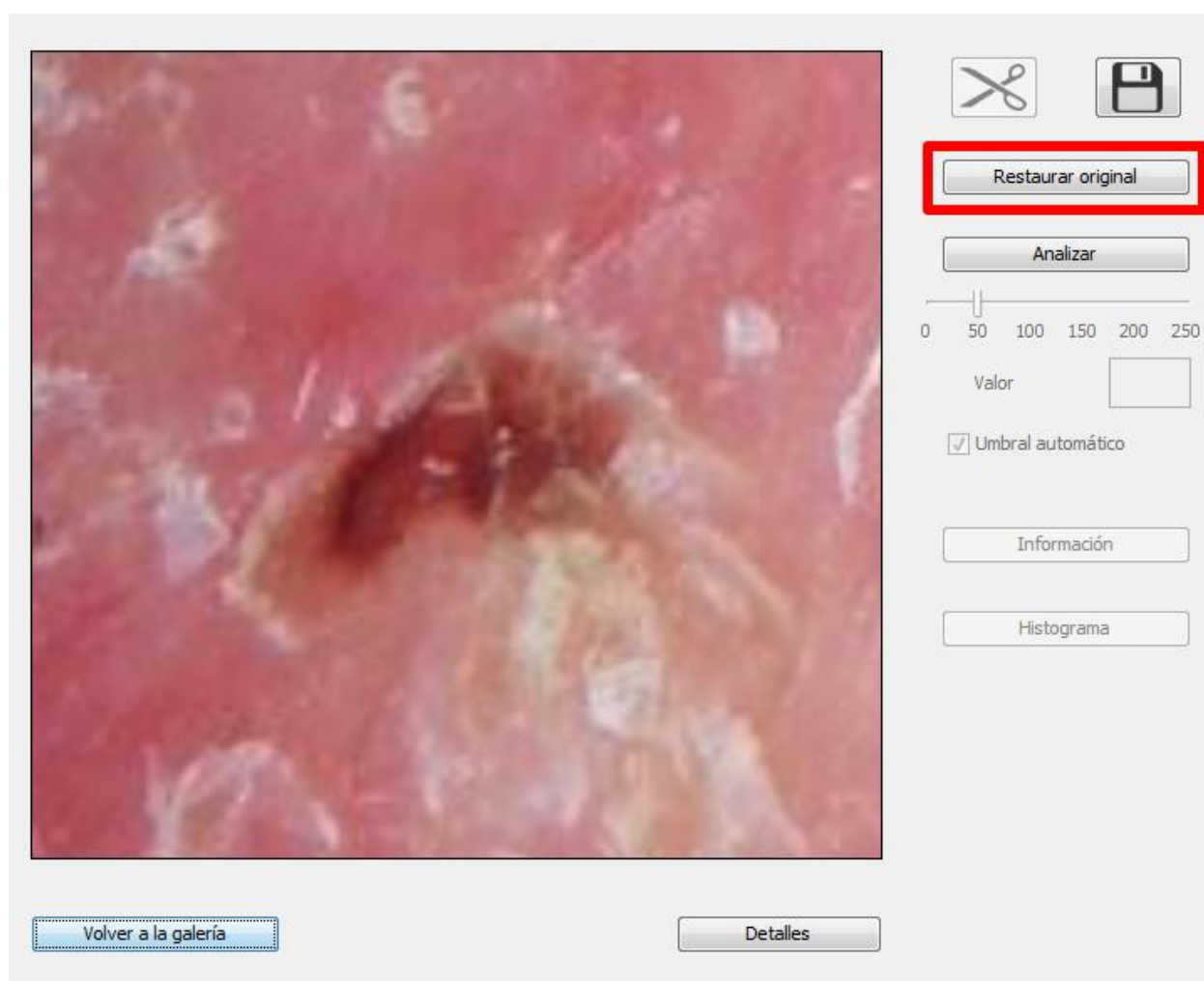


Figura A11: Restaurar Original

A2.2. Procesar una imagen

Requerimientos iniciales:

- Situarse en la ventana de análisis de imagen.

El primer paso para analizar la imagen consiste en **procesarla**, esto es, identificar la piel con problemas y diferenciarla de la zona de piel carente de interés. Para comenzar este proceso, desde la **ventana de análisis**, pulsar “**Analizar**” (figura A12). Bajo esta operación es posible recortar la imagen previamente si se considera necesario.



Figura A12: Analizar imagen

Observará que la imagen se ha transformado, apareciendo ahora dos regiones en **blanco y negro** (*figura A13*). Idealmente, la zona negra es la parte de piel considerada de interés, esto es **afectada**, y la zona blanca es piel **sana**. Esta apreciación se obtiene de forma automática, gracias al **valor de umbral**. Si se considera que la zona mostrada no ha sido bien localizada o puede ser susceptible de una mejor localización, se puede cancelar la opción “**Umbral automático**”. A continuación es posible desplazar el deslizador o introducir manualmente el nuevo valor de umbral (*figura A14*). Para capturar más piel carente de interés, se debe seleccionar un valor de umbral menor, mientras que si quiere capturar más piel afectada, el valor debe ser mayor. Esto también dependerá de las propiedades de color de las zonas afectadas, de forma que podrían invertirse los papeles en función de esta circunstancia.

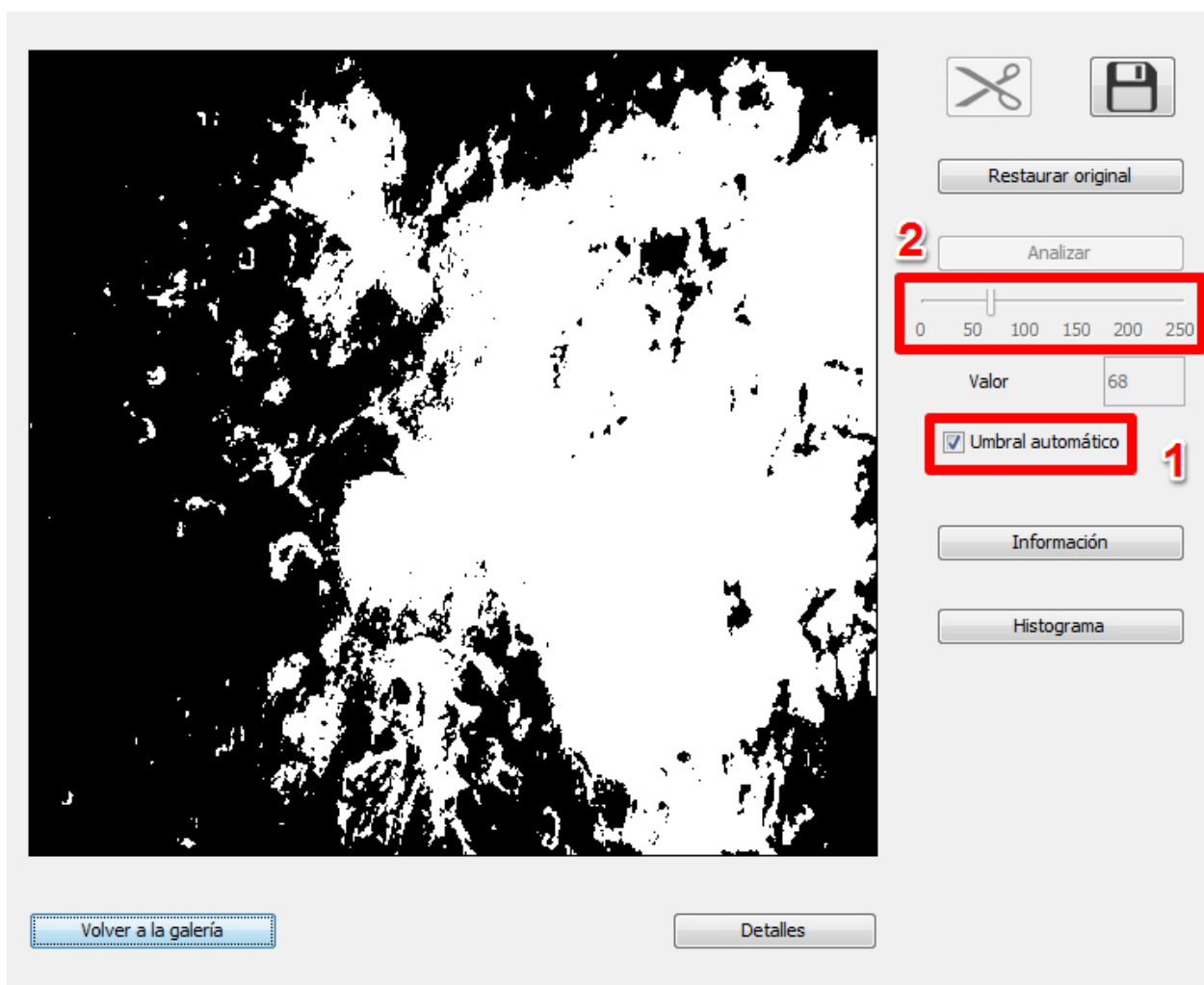


Figura A13: Imagen analizada, umbral automático

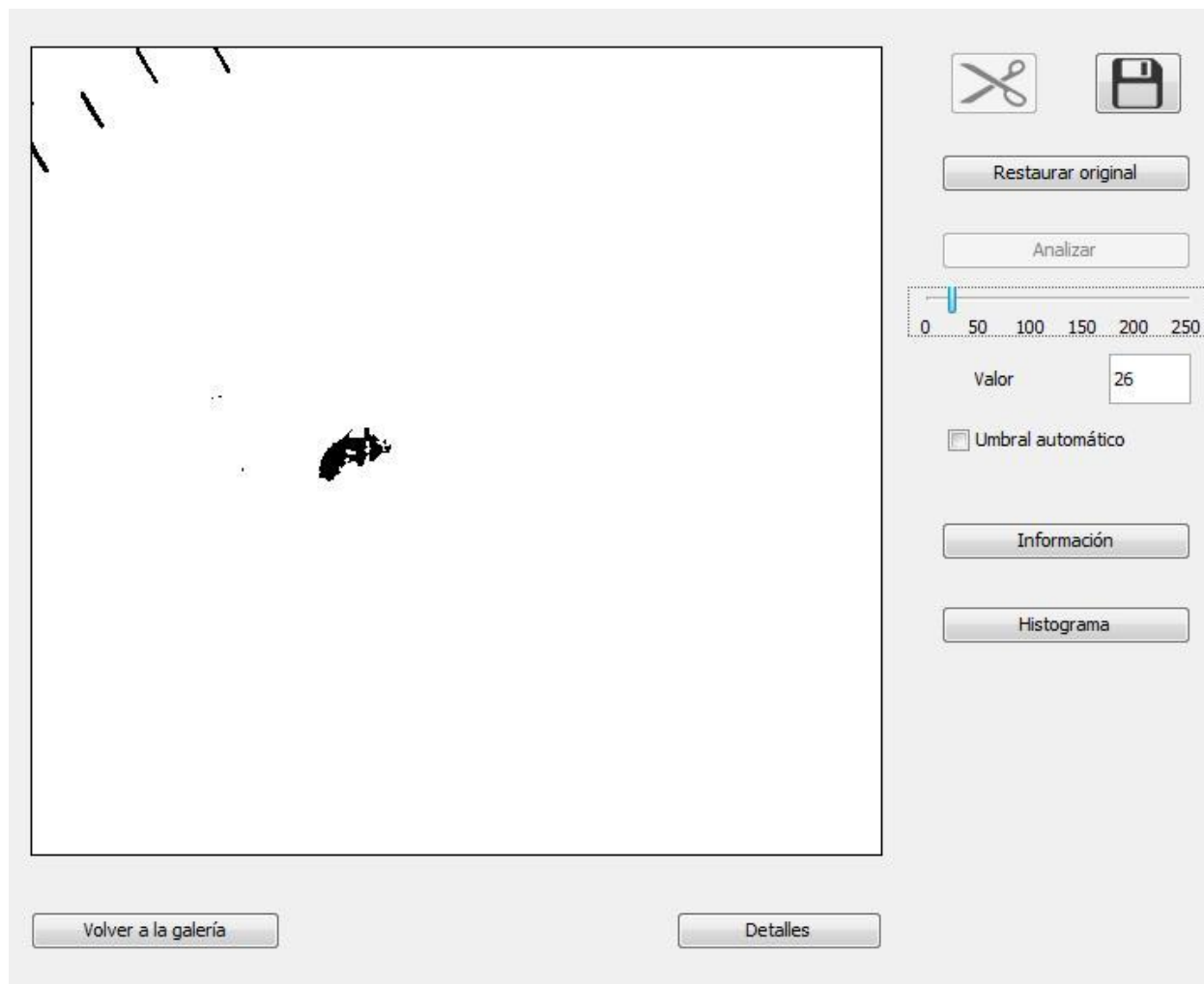


Figura A14: Imagen analizada, umbral manual

Cuando la zona a identificar sea la deseada, la imagen estará lista para ser guardada, obtener su **información** o visualizar el **histograma**. Si no se está satisfecho y se quiere variar parte del proceso, se puede pulsar **“Restaurar original”** para comenzar de nuevo.

A2.3. Obtener información de la imagen

Requerimientos iniciales:

- Disponer de una imagen analizada

Con la imagen analizada en pantalla, pulsar **“Información”** (figura A15).

Una vez pulsado dicho botón, aparecerá una nueva ventana donde se mostrará la información de la imagen analizada (*figura A16*).

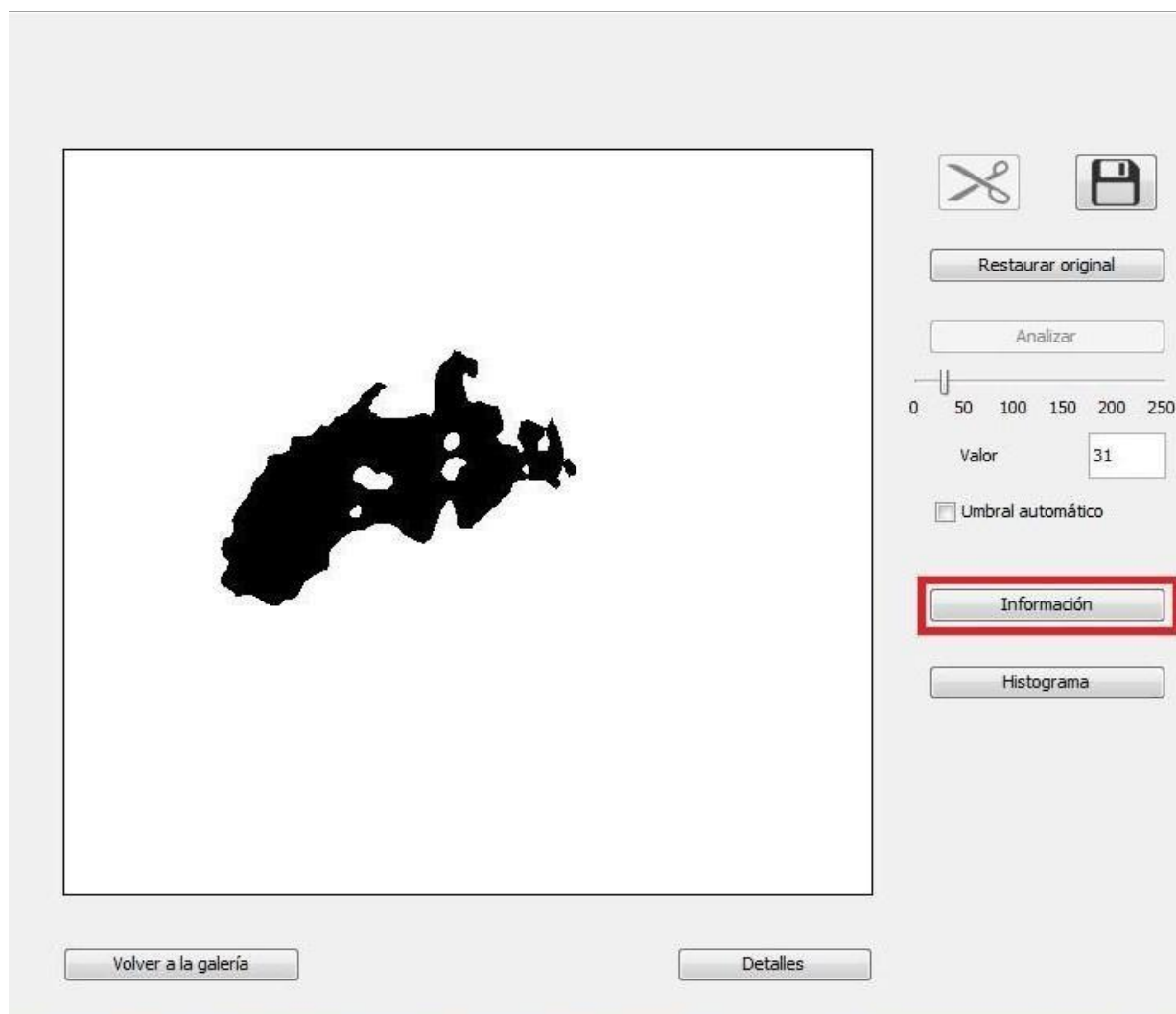


Figura A15: Obtener información

Información de la imagen			
	Rojo	Verde	Azul
Media	178	98	102
Varianza	376	491	563
Asimetría	-12054.0	26193.0	28235.0
Energía	7.0	2.0	2.0
Entropía	-1.7812270...	-9.812055...	-1.5852475...
Área de la forma: 133312 píxeles			

Figura A16: Información de imagen

A2.4. Histograma de la imagen

Requerimientos iniciales:

- Tener una imagen analizada

Con la imagen analizada en pantalla, pulsar **“Histograma”** (figura A17).

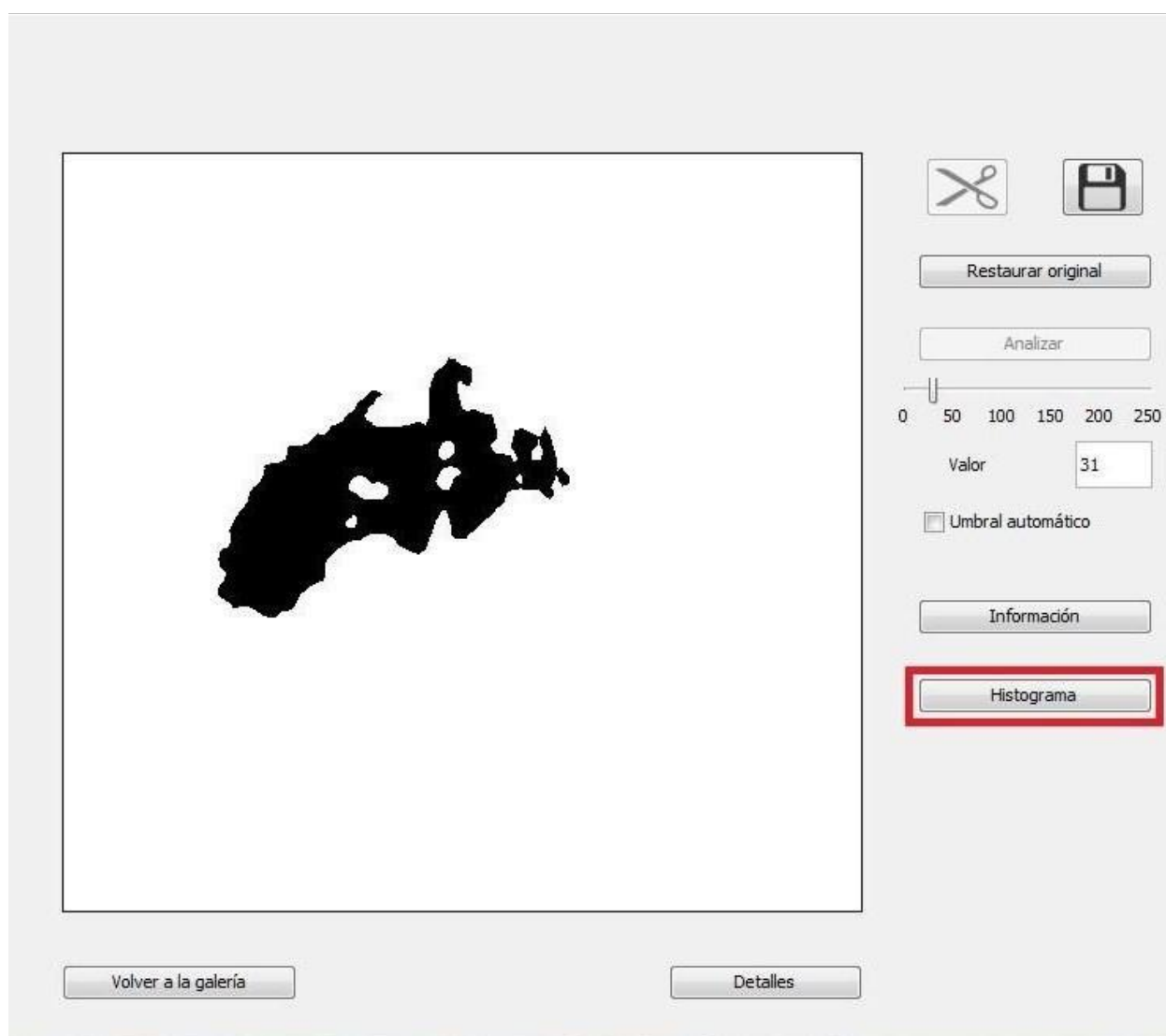


Figura A17: Obtener histograma

Una vez pulsado dicho botón, aparecerá una nueva ventana donde se mostrará el histograma de la imagen analizada (figura A18).

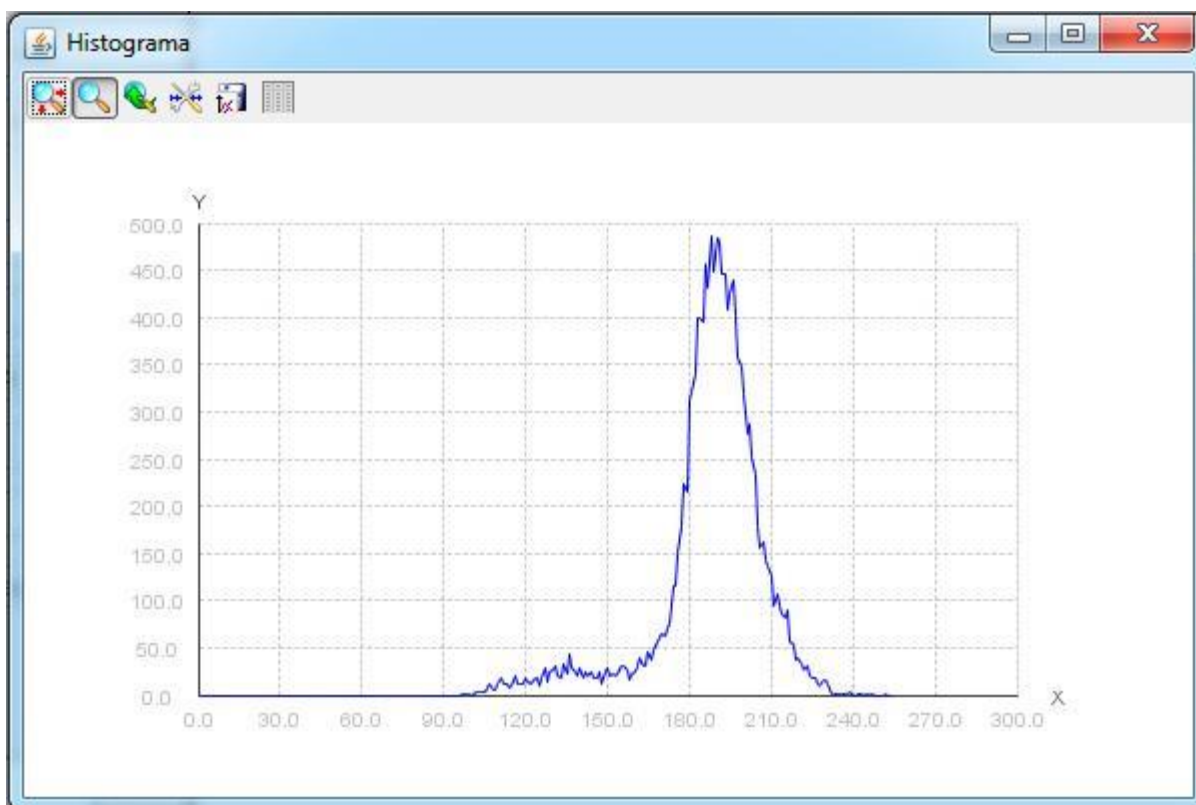


Figura A18: Histograma de imagen

A2.5. Guardar imagen

Requerimientos iniciales:

- Disponer de una imagen analizada o recortada

Es posible guardar la imagen siempre que se realice algún cambio sobre ella, ya sea un recorte o su procesamiento. Para realizar la operación de guardado, debemos pulsar en el botón de **“Guardar”**, representado con el icono de un disquete (*figura A19*).

Una vez pulsado dicho botón, aparecerá un cuadro de diálogo que solicitará introducir el nombre de la imagen. Debemos introducir el nombre y pulsar **“Aceptar”** (*figura A20*).

Cuando aceptemos el nombre, aparecerá otro cuadro de diálogo para introducir las anotaciones deseadas en relación a la imagen. Finalizada esta operación se pulsa **“Aceptar”** (*figura A21*).

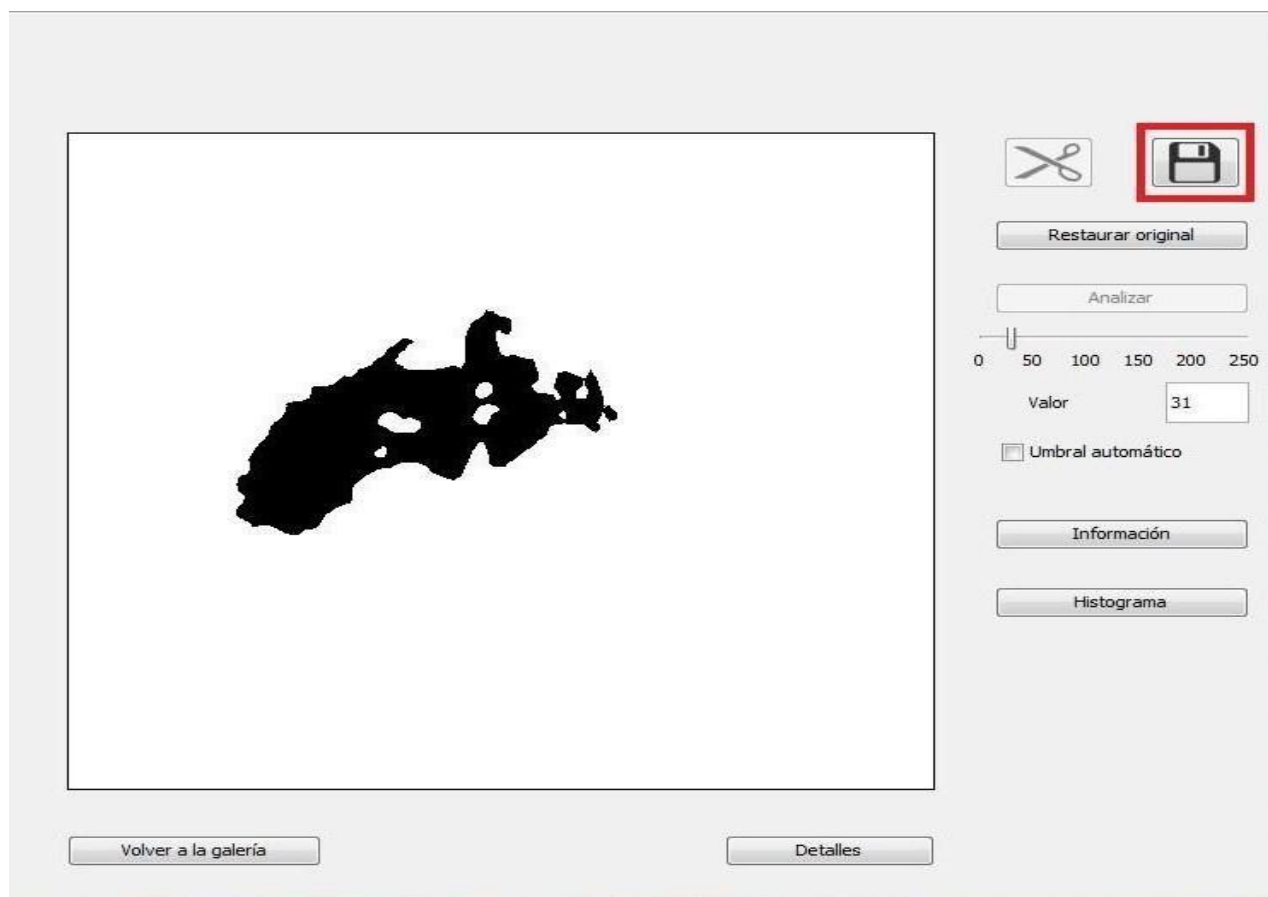


Figura A19: Guardar imagen

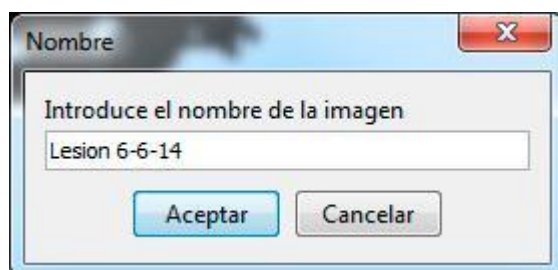


Figura A20: Introducir nombre de imagen

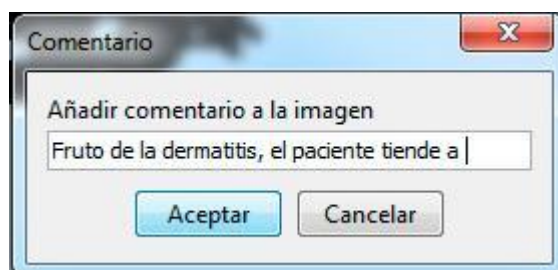


Figura A21: Introducir comentario de imagen

Finalmente, se mostrará un cuadro en el que se indicará que la imagen se ha guardado correctamente. Para finalizar el proceso, es necesario pulsar “**Aceptar**” (figura A22).

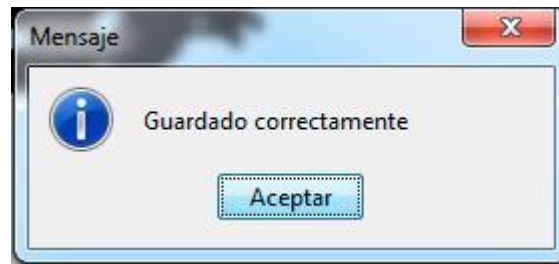


Figura A22: Guardado correctamente

A3. Historial imágenes tratadas

Requerimientos iniciales:

- Tener acceso a la aplicación con permisos de usuario médico
- Tener un paciente seleccionado.
- El paciente tiene que tener alguna imagen en su historial.

Conviene reseñar que el acceso al gestor de pacientes con permisos de administrador está restringido.

El siguiente paso es común para todos los puntos siguientes: si del paciente seleccionado no se dispone de imágenes tratadas del conjunto que conforman su historial de imágenes sólo se mostrará la imagen original sin tratar.

Para posicionarse en esta ventana, desde la perspectiva de **Historial de Imágenes** seleccionamos una imagen y una vez que ésta se encuentra seleccionada pulsamos en “**Imágenes relacionadas**”.

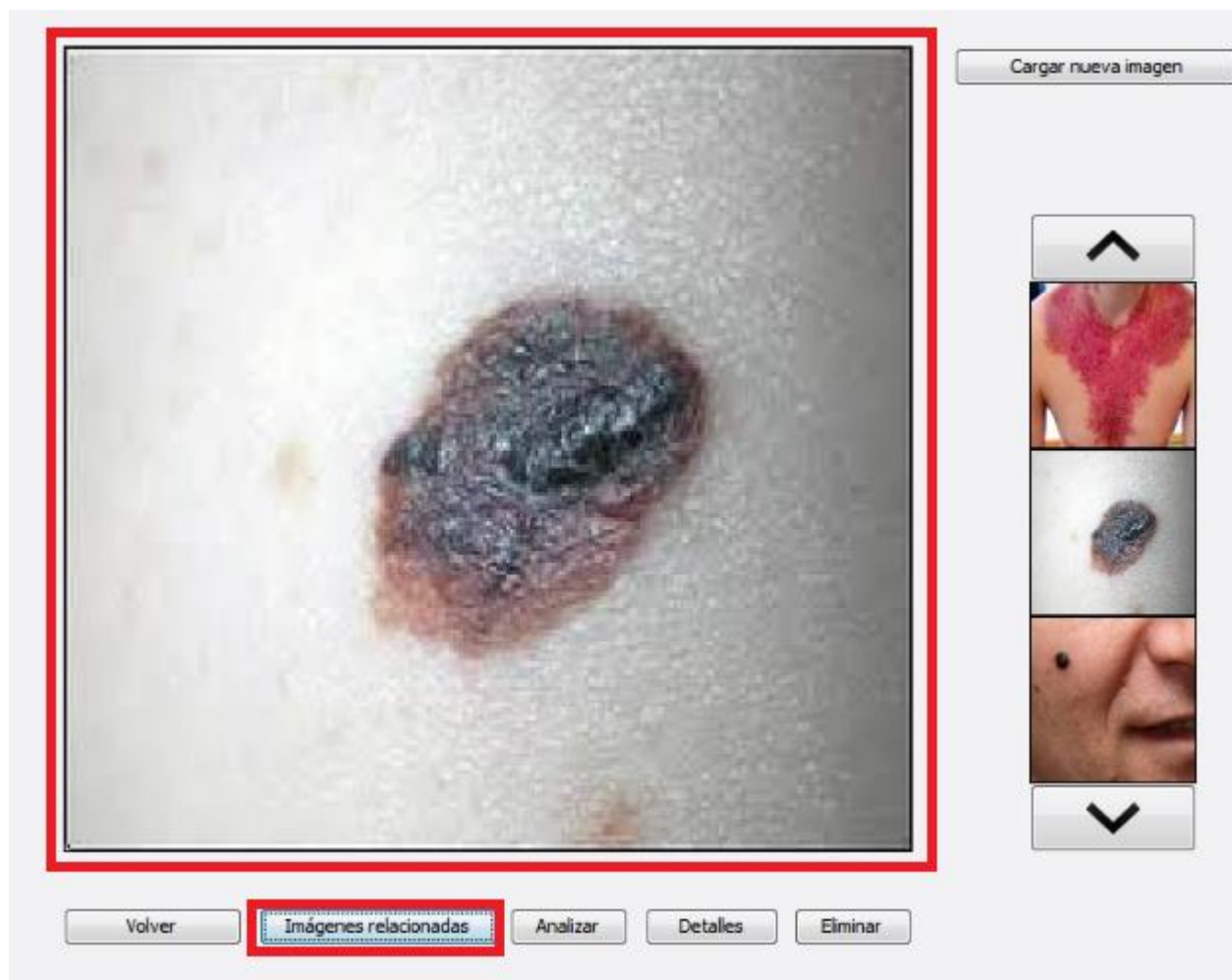


Figura A23: Historial de imágenes relacionadas

A3.1. Volver al historial de imágenes.

Una vez situados en esta ventana **Imágenes relacionadas** pulsar en el botón “**Volver a galería**” (figura A24).

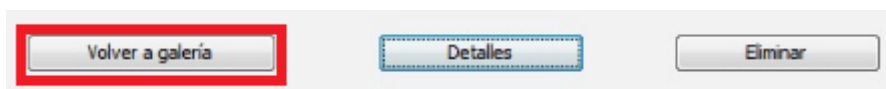


Figura A24: Volver a galería

Con esta acción volvemos a la ventana principal de **Historial de Imágenes** del paciente.

A3.2. Eliminar imagen tratada.

Seleccionamos la imagen tratada que deseamos eliminar y pulsamos el botón **“Borrar”** (figura A25).

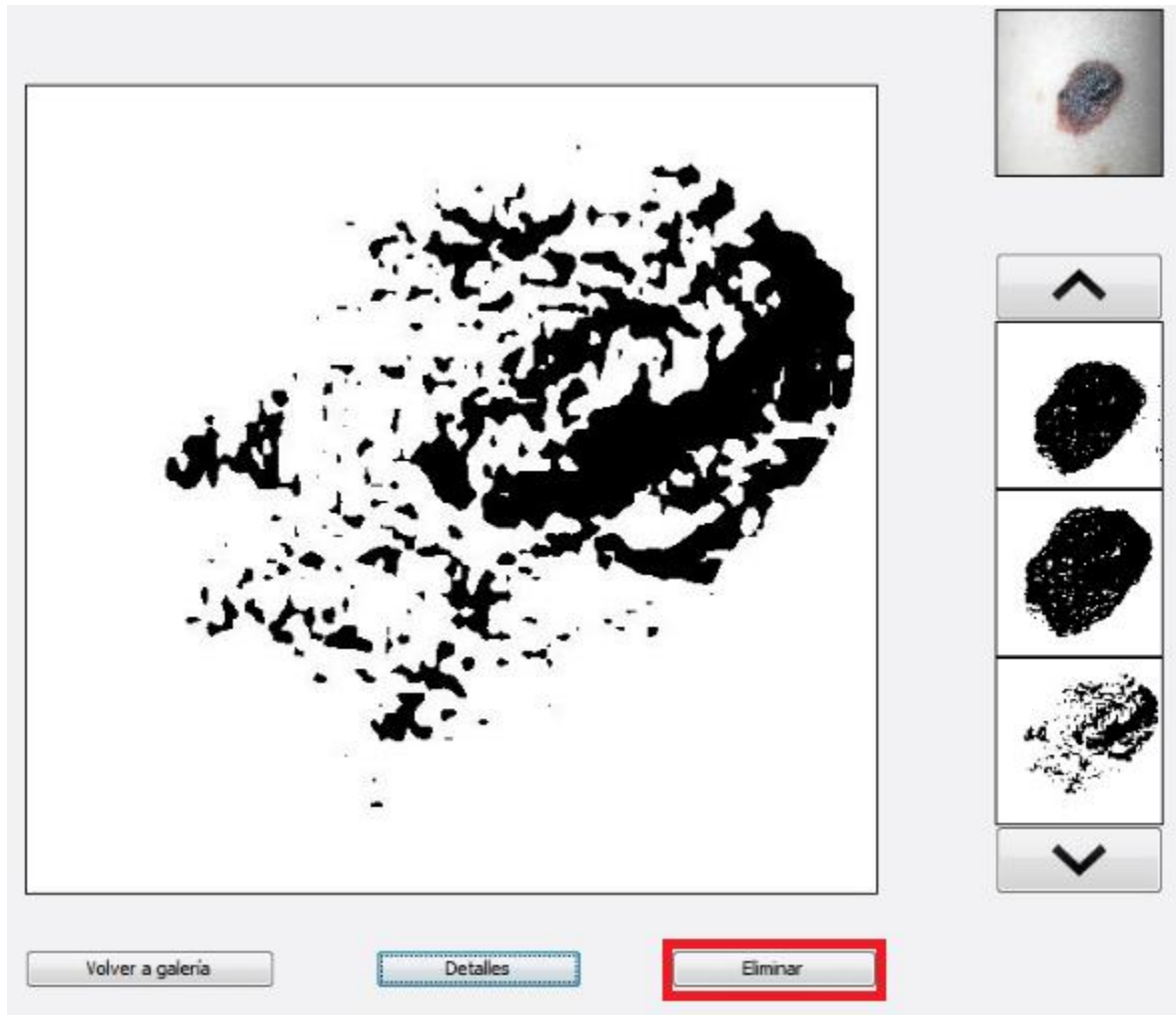


Figura A25: Borrar imagen tratada

A3.3. Consultar detalles de imagen tratada.

Para ejecutar esta consulta pulsamos seleccionar las imágenes de las que queremos ver sus comentarios introducidos previamente por el doctor y pulsamos **“Detalles”** (figura A26).

Por último aparecerá una ventana emergente donde el doctor podrá visualizar todos los comentarios disponibles en relación con esa imagen tratada, permitiendo también la edición de los mismos junto con las operaciones de tratamiento de imágenes aplicadas.

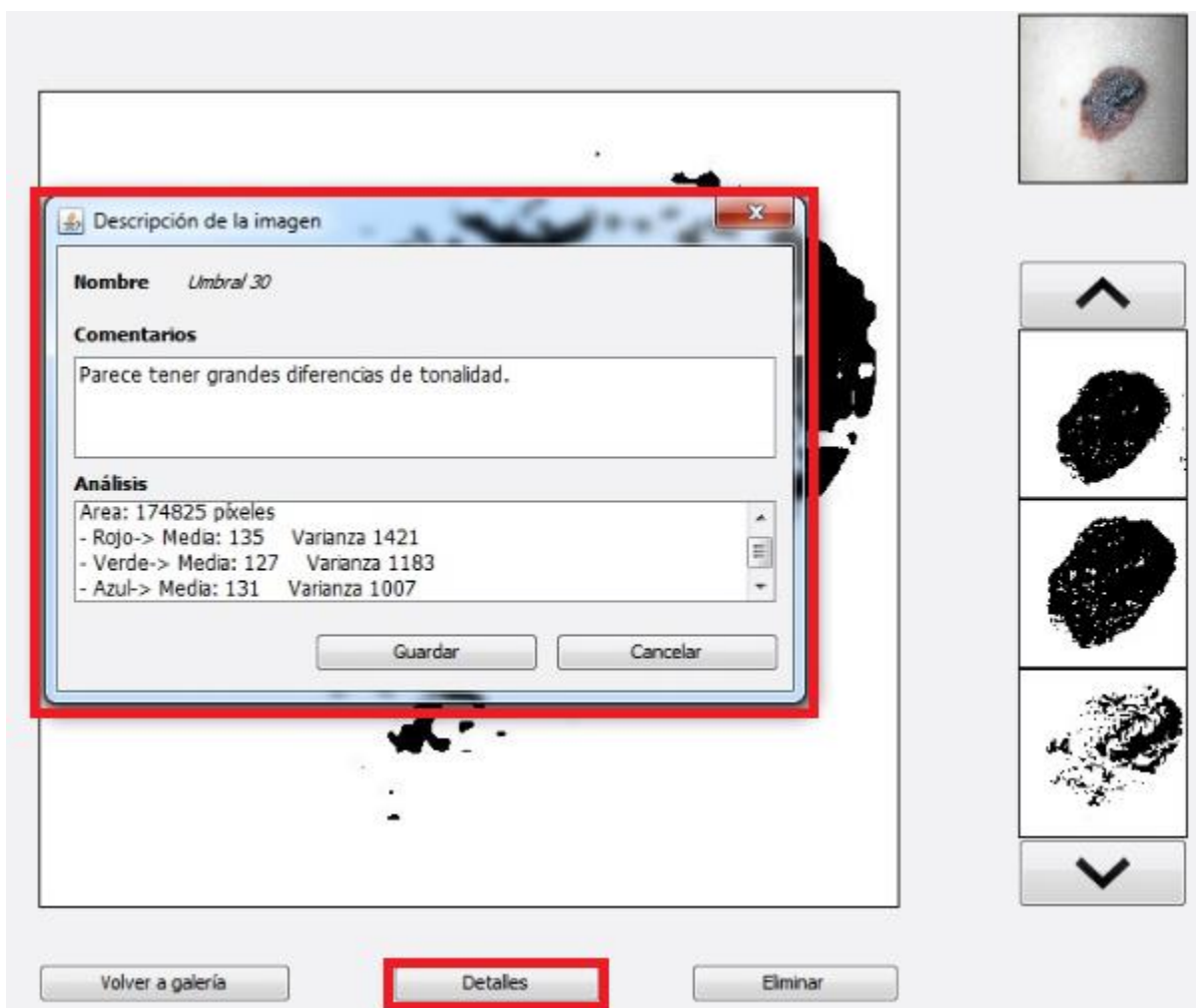


Figura A26: Detalles de imagen tratada